DOCTOR OF SCIENCES

Aligning Access Rights to Governance Needs with the Responsability MetaModel (ReMMo) in the Frame of Enterprise Architecture

Feltus, Christophe

Award date:
2014

Awarding institution:
University of Namur

A thesis submitted for the degree of
Doctor of Science (Computer Science)

Aligning Access Rights to Governance Needs with the Responsibility MetaModel (ReMMo) in the Frame of Enterprise Architecture

Christophe FELTUS

Faculty of Computer Science, University of Namur, Belgium
Public Research Centre Henri Tudor, Luxembourg

Co–Supervisors:
Prof. Dr Michaël Petit
Prof. Dr Eric Dubois
Defended in March, 2014

Graphisme de couverture : © Presses universitaires de Namur

Imprimé en Belgique
ISBN : 978-2-87037-843-4
Dépôt légal: D / 2014 / 1881 / 34

I dedicate this thesis to those who are my unfailing source of motivation, my wonderful children Kha Luân, Luu Ly and Y Lan

Acknowledgements

I am heartedly thankful to Prof. Michaël Petit for accepting to co–supervise my PhD. Michaël has always been available to accompany me throughout my research. His scientific advice and rigour in the field of modelling were essential to the completion of this thesis. It is during interesting discussions, at the Faculty of Namur, that lots of ideas have emerged and have been refined. Michaël has been constantly full of enthusiasm and was a valuable support for me.

I warmly thank Prof. Eric Dubois, co–supervisor of my PhD, who allowed me to pursue the research at the Public Research Centre Henri Tudor. Due to his sharp vision in the field of governance and information systems, Eric has guided me through crucial and promising research areas. His door was always open to share his huge expertise and experience. This sign of trust has been an extremely useful support.

I also thank the members of my thesis Steering Committee: Prof. Yves Pigneur, Prof. Claire Lobet–Maris and Prof. Moris Sloman. It has been an honour to be advised by them and to enhance my research according to their helpful comments.

My sincerest gratitude also goes to the members of my Jury: Prof. Najid Habra, Prof. Jean–Noël Colin, Prof. Claire Lobet–Maris, Prof. Yves Pigneur, and Prof. Hend*erik* A. Proper. All have kindly accepted the task of judging the results of my work.

Without a good case study, it would have been impossible to review the advantages of the Responsibility metamodel and its integration with ArchiMate and RBAC. The Centre Hospitalier de Luxembourg and the European Court of Auditors have accepted the challenge to be the basis of this evaluation. All my thanks go to both of those institutions.

The commitment of Patrick Recht has allowed me to access an impressive amount of pragmatic and relevant information. Thanks to Patrick, I was connected with many very interesting peoples from the hospital staff. In particular Frank Schmitz, Laurent Wehr and Marco Pappafava who have allowed me to acquire accurate knowledge regarding the management of access rights in the healthcare sector.

I would also like to thank Prof. François Vernadat with regards to the time he spends to depict the process of User Provisioning and User Account Management at the European Court of Auditors. François has provided me with much advice along all the steps of the case study and has been a guide in the field of identity management.

Special thanks to Iver Band (from the Standard Insurance Company) for having helped me in preparing The Open Group workshop, for sharing his knowledge in the

field of enterprise architecture and for having made me discover every interesting nook and corner of San Francisco. I also thank Prof. Steven De Haes and Prof. Wim Van Grembergen, from the University of Antwerpen, who provided my with essential advice and validation related to COBIT.

It would be unforgivable to forget my colleagues for their helpful discussions. I especially thank André Rifaut, who has been present from the very beginning of my research, for providing me with a plethora of essential recommendations, Dr Khaled Gaaloul for having reviewed the results related to ArchiMate, Eric Grandry for the talks we had regarding the mapping of the Responsibility metamodel and ArchiMate, and Damien Nicolas for helping me with my numbering issues with LaTeX. I want to particularly thank Margot Hartman for having conscientiously checked all my English texts in detail.

Finally, I would like to cordially thank my friends, my family and my parents who always support me in whatever I pursue, and most of all, my wife Kinh Trang who constantly comforts and encourages me, especially during the final stages of this thesis. *Thanks to all!*

Members of the Jury

Prof. Dr Najid Habra,
Daen of the Faculty of Computer Science,
University of Namur, Belgium

Prof. Dr Michaël Petit
Thesis Co–Supervisor,
University of Namur, Belgium

Prof. Dr Eric Dubois,
Thesis Co–Supervisor,
Public Research Centre Henri Tudor, Luxembourg

Prof. Dr Jean–Noël Colin,
University of Namur, Belgium

Prof. Dr Claire Lobet–Maris,
University of Namur, Belgium

Prof. Dr Yves Pigneur,
University of Lausanne, Switzerland

Prof. Dr Hend*erik* A. Proper,
Public Research Centre Henri Tudor, Luxembourg
Radboud University Nijmegen, the Netherlands

Members of the Steering Committee

Prof. Dr Michaël Petit
Thesis Co–Supervisor,
University of Namur, Belgium

Prof. Dr Eric Dubois,
Thesis Co–Supervisor,
Public Research Centre Henri Tudor, Luxembourg

Prof. Dr Claire Lobet–Maris,
University of Namur, Belgium

Prof. Dr Yves Pigneur,
University of Lausanne, Switzerland

Prof. Dr Morris Sloman,
Imperial College London, United Kingdom

Abstract

Nowadays the economy relies on companies evolving in an increasingly highly regulated environment, having their operations strongly formalised and controlled, and being often organised following a bureaucratic approach. In such a context, aligning the business operations with the appropriate IT infrastructure is a challenging and critical activity. Without efficient business/IT alignment, these companies face the risk not to be able to deliver their business services satisfactorily and that their image is seriously altered and jeopardised. Among the many challenges of business/IT alignment is the access rights management which should be conducted considering the rising governance needs, such as taking into account the business actors' responsibility. Unfortunately, in this domain, we have observed that no solution, model and method, fully considers and integrates the new needs yet. Therefore, the thesis proposes firstly to define an expressive Responsibility metamodel, named ReMMo, which allows representing the existing responsibilities at the business layer and, thereby, allows engineering the access rights required to perform these responsibilities, at the application layer. Secondly, the Responsibility metamodel has been integrated with ArchiMate to enhance its usability and benefits from the enterprise architecture formalism. Finally, a method has been proposed to define the access rights more accurately, considering the alignment of ReMMo and RBAC. The research was realised following a design science and action design based research method and the results have been evaluated through an extended case study at the Centre Hospitalier de Luxembourg.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 New challenges for information systems and access rights management

This is nowadays recognised by all and commonly agreed upon that the emergence and the growing up of the information system (IS) has revolutionised the way we communicate with each other and the manner in which we do business. In parallel to this growth, the information system has opened the door, for people in private or professional capacity, to an immeasurable source of information and it has permitted to perform operations never imagined before. But, the evolution of the information system is not yet completed and the tendencies for the next decades appear already to be well–known: more openness, more interconnectivity and real–time interactions, and more heterogeneity (Milanovic et al. (2009)). If we depict in depth these trends of the information system's evolution, we also observe that its management is more and more outsourced to external companies (Huws et al. (2004)) and that, on the other hand, the professionals tend to keep concentrated on their core businesses. These trends have been advanced with the emergence of new technologies like, for instance, the distributed and cloud computing that, since 2000, tends to distribute information technology applications and services on remote servers (Birman et al. (2009)).

In this context, it is manifest that business companies must set up security mechanisms to carefully manage the information in their possession. Without such mechanisms, and without an efficient control over their information system, they will be unable to survive more than a couple of hours (Peppard (2004), Spremić (2011)) and will suffer more or less important financial impacts (Garg et al. (2003)). This is especially true for companies that are intensely present on the Internet and that have their turnover through e–Business activities.

Among these security mechanisms to be set up, the deployment of a structured security management framework is fundamental, to guarantee the availability, the integrity, the confidentiality of the information, as well as the accountability of the employees who access it. The first challenge in this area stays in storing and archiving a mass of continuously growing data and the second challenge stays in making those data (i) available at any time (Bhagwan et al. (2005)), (ii) to all kinds of stakeholders such as, for instance, the access to a healthcare system by the clinicians, patients, and all different healthcare specialists (Goldberg et al.

(2011)) and (iii) on many types of media such as the mobile devices which are omnipresent, for business applications as well as for entertainment or personal duties (Holzer and Ondrus (2009)). Finally, they need to stay compliant with the applying regulations, related to a country (Stieghahn and Engel (2010)) or to a professional sector such as IT Governance Institute (2007).

At the information system level, the management of the access rights and their alignment with the business activities appear of crucial importance. Many mechanisms have been proposed for two decades, to elaborate adequate models in this field of access rights. Some of them have appeared to be commonly admitted, like the Role Based Access Control (RBAC) (Ferraiolo et al. (2001), that has emerged in 1996 as a reference model in this discipline. Indeed, in many companies, the management of employee's permissions and rights is done by using the central concept of role which permits to manage a large amount of users, on one hand, and the permissions assigned to the role, on the other hand. This increasing of RBAC usage is illustrated in the analysis from the 2010 Economic Analysis of Role Based Access Control Final report (O'Connor and Loomi (2011)) that shows, as highlighted in Figure 1.1, that the use of roles in American companies with more than 500 employees has significantly grown since 1994. Indeed, the number of employees that have their permissions managed using roles has increased from 2.5 percent in 1995 up to 40.5 percent in 2009. Additionally, more than 84 percent of them agree that the use of roles improved the efficiency of maintaining the organisation's access control policy.

Although at a technical point of view, many of such access control models exist, approaches and methods to instantiate them considering business input are still missing. This lack of solution is often the origin of access rights provisioning which are not the most accurate nor stringent to the employee having accountabilities and responsibilities for business tasks. The accuracy and strict alignment between these business tasks and the corresponding needed access rights is actually formally requested by governance standard and norms, as explained in Section 1.2, which require, for instance, to respect the principle of least privilege or the separation of duties. Although enterprise architecture modelling has appeared to be a powerful tools to model those concepts from the business and the application layers, as well as the association between them, rigorous alignment methods are missing, and needs of improvements also exist in this field, as afterwards explained in Section 1.3.

1.2 Needs for governance

Exposed to these fast mutations in the way they make business and to face these new challenges, the actors of the economic world are continuously asked to review their business processes to align them with the arising professional modifications. The information system that sustains those processes is, consequently, also pressed to continuously be adjusted with the process modifications. To support the fast growth of the economy, the corporate governance has emerged in the eighties as a discipline that aims to frame all the aspects necessary to understand the new issues fostered by the arising business opportunities. It is, therefore, different to the classical management that focuses on the day to day follow up of those activities. This corporate governance has been absorbed, little by little, across all the company layers of activities, so that we have seen the appearance of specific rules and needs for the governance of the project, governance of the customer relationships, governance of the production, governance of the security,

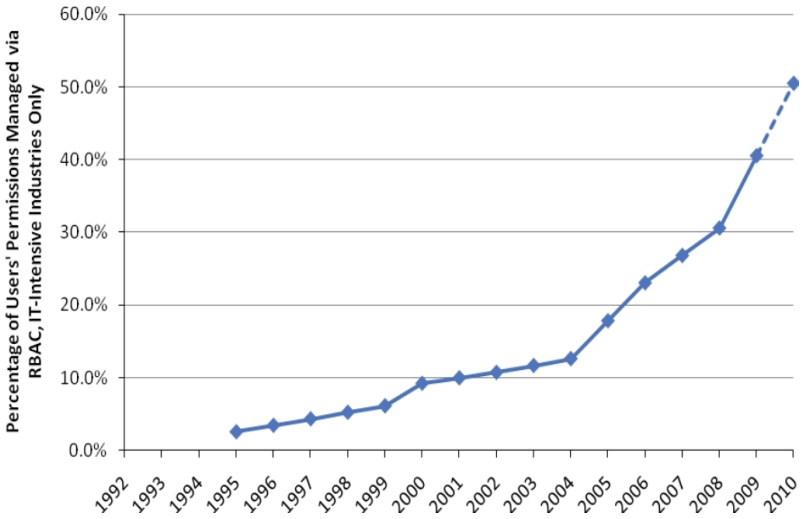


Figure 1.1: RBAC deployment until 2010, **Source:** O'Connor and Loomi (2011)

and so forth. New needs have progressively arisen like, for instance, the need to have employees responsible for the business tasks they are accountable for, to have them committed to their obligations, and to have them answerable for the results.

The field of IT has not been set apart of these modifications and improvements dictated by the emerging needs. Information technology governance has appeared, for the last 10 years, to be an important matter to be handled by the board of directors as well as the IT managers. Since then, academic surveys (MIT (2002)), as well as industrial analyses, have highlighted the need to enhance the governance of Information Technology (IT), such as the control, the risk management, the business/IT alignment and the management of the access rights. All of these domains are gathered under the Corporate Governance of the IT umbrella and are progressively integrated in standards and norms such as ISO38500 (2008) that provides principles for the corporate governance of IT, COSO (2004), the voluntary private–sector organisation that has established an internal control model that allows companies to assess their control systems, SOX (2002) that describes the needs and specific mandates for financial reporting, or Basel2 (2004) that defines rigorous risk and capital management needs for the banking sector.

1.3 Needs for enterprise architecture

To develop the information system, engineers need to define methods and techniques to align, as far as possible, this system with the processes that they support. This alignment has for objective to improve the definition and the deployment of the information system at all layers, from the business layer where the strategy of the information system is elaborated, down to the application layer where the production activities are supported by dedicated and well tailored IT applications Lenz and Kuhn (2003) and (Tan and Gallupe (2006)). Through their *Strategic alignment model*, Henderson and Venkatraman (1993) had proposed a four alignment perspectives method to connect the business strategy with the IS infrastructure and processes (Figure

1.2). The *technology potential view* (1) considers the business strategy as the incentive both for designing the IT strategy, firstly, and for elaborating the IS infrastructure and processes, secondly. The *strategy execution view* (2) considers the business strategy as the incentive for the organisational infrastructure and processes design, and for the IS infrastructure and processes elaboration. The *competitive potential view* (3) illustrates the case where new IT opportunities generate new business services or products and thereby, influence the business strategy and, as a result, the organisational infrastructure and processes. The *service level view* (4), as the competitive potential view, is also driven by IT opportunity which necessitates fast changes of the IT infrastructure and processes that support the end users interest.

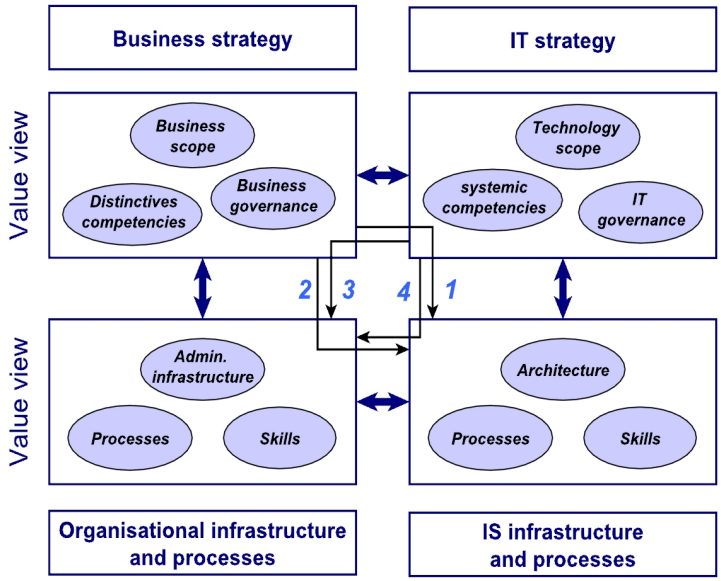


Figure 1.2: Strategic Alignment Model (SAM), **Adapted from:** Henderson and Venkatraman (1993)

Enterprise architecture reference models such as TOGAF (Lankhorst and van Drunen (2007)) are types of tools especially developed to contribute in supporting this alignment. Enterprise architecture is a technique used to give businesses and IT static views of the corporate architecture as well as of the links between those views. The advantage of the enterprise architecture models is that they propose means to model and better understand the enterprise, the interconnections and interdependency between the processes, the people and the systems. Consequently, they permit to reduce the complexity and allow better decision–making.

The activities represented by enterprise architectures are, traditionally, business (or core) activities and answer the question *What to do?* According to Liebwein (2006), activities may also be structural activities and answer the question *How to do it?* The goal of the latter is to support the realisation of the business activities according to different perspectives such as the quality, the governance or the security. As explained in Figure 1.3, the structural activities aim at creating supportive value for the business. Therefore, these activities may also be formalised by processes assigned to employees and supported by applications and dedicated infrastructures, and thereby, modelled with enterprise architecture model.

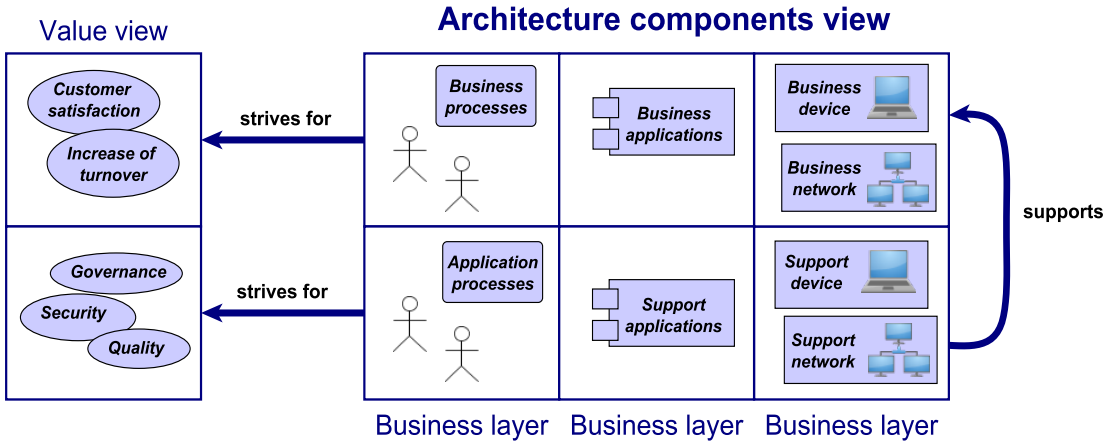


Figure 1.3: Overview of the enterprise architecture layers

Despite the advantages conferred by those business architecture models, we observe in the latter a lack of presence of concepts dedicated to the modelling of access rights management, considered as part of structural governance activities. In ArchiMate® (Lankhorst (2004)), for instance, an *assignment* association exists between the *business role* and the *business process*, but the enterprise architecture model does not explain why such association exists, neither what it implies in terms of the rights to be assigned to the *business role*.

1.4 The research problem domain

Our research problem domain is related to the access rights management and, more particularly, to the enhancement of the definition of these rights in the frame of governance and to their implementation through enterprise architecture frameworks. This research is focussed on those frameworks and, therefore, aims to improve the links between the concepts from the business layer and those from the application layer. Figure 1.4 provides an overview of the components which are involved in access rights management. Nowadays, the application layer (**Item 5**) is defined according to the business layer (**Item 6**) using requirement engineering methods (**Item 8**). The rights assigned to the business users regarding application components at the application layer are formalised in the access rights policies (**Item 7**) which are constructed with rights engineering methods (**Item 3**). To define the rights, these rights engineering methods consider the employees' requirements related to the use of the information system, to perform the tasks they are assigned to. As a consequence, these requirements also highlight which information needs to be accessed by which employee. In addition, the access rights policies are formalised following the access control model's specifications (**Item 4**) which model the access rights, considering a set of organisational artefacts (**Item 6**). Those artefacts are, among others the tasks, the business processes, the employees, the roles, and sometimes the hierarchy between the roles. This is the case, for instance, of RBAC (Ferraiolo et al. (2001)) that exploits the organisational artefacts of a *role* defined as a *job function within the context of an organisation with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role* to provide access rights to the employees.

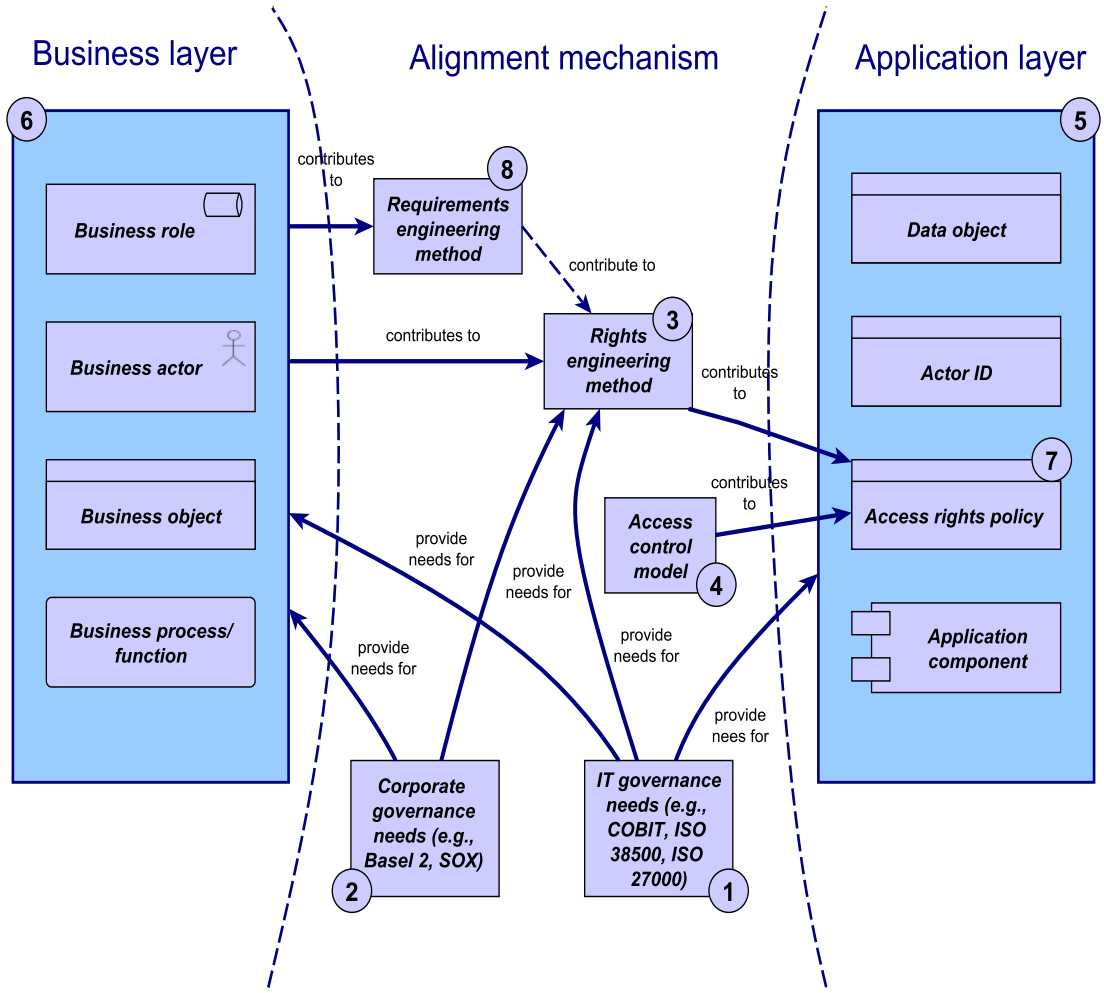


Figure 1.4: Access rights management components

The arising of the corporate governance standards and norms (**Item 2**), or the information technology governance standards and norms, in particular (**Item 1**), provide new needs related to the alignment of the business layer (**Item 6**) with the application layer (**Item 5**), as well as on the rights engineering methods (**Item 3**). In practice, however, there is a miss of consideration of these governance needs. We observe that the access control models and rights engineering methods still remain very technical and that the organisational artefacts (**Item 6**) are misaligned with, and are sparsely integrated in the access control models (**Item 4**). The past researches, further detailed in the state of the art in Chapter 2, were mostly fulfilled with the objective of performing rights engineering without taking into consideration these arising needs, like the request of formalising the employees' responsibilities and accountabilities related to a business task.

Due to this lack of consideration of the governance's needs during the modelling and the engineering of the employees' access rights, we have observed the two following problems through both of our case studies in existing companies:

1. The rights are most often assigned to employees because they are assigned to one or more business role(s) rather than to their real responsibilities. In the best cases, the roles are assigned to the performance of a set of tasks but it is not systematic. In practise, we observe that, at the business layer, a task often needs the intervention of more than one employee each with different responsibilities. Some of them have the obligation to do the task, others have the obligation to achieve the goal of the task, others to supervise it, to make decisions, to approve its realisation and so forth. Each of these obligations does not require the same access rights. I.e., the employee that approves the realisation does not require the same rights as the person who performs it. Current access rights management models do not consider the responsibility of the employee according to what they really have to do with regards to a business task.

   This problem was observed, among others, in a European public administration of about one thousand employees. In this administration, many employees were assigned to the role "IT administrator", although, many may realise different tasks and have different responsibilities related to these tasks. For instance, some of them take care of the Novell system, others of the Windows servers, others of the mail application, and so forth. Additionally, if we only consider the management of the users account related to the Novell system, we also encounter employees that really manage the user accounts, others that are accountable for always having the user accounts suitably defined, and so forth. In this case, the problem is that some employees are assigned to the role "IT administrator" and thus receive too many rights according to the tasks they have to perform and other employees are not assigned to the "IT administrator" role although they are involved in it, like the employee which is responsible for the accuracy of the rights. In the first case, there is a security problem. Indeed, providing the employees with more rights that they really need is increasing the threat of unauthorised accesses. In the second case, there is a problem of poor performance of the employees who do not receive all the information needed to check the accuracy.

2. The second problem is that a company has to decide to assign an employee either (1) to a unique role which merges the business role (as defined in the employment contract or job profile) and to the "Application" role defined at the application layer such as it is used in RBAC (Section 2.2.3), or (2) separately to both roles.

   With the first, the company tends to assimilate both roles although these roles have distinct objectives. The aim of the business role is to gather employees having specific business tasks to perform, in a particular organisational context such as the hierarchy (Barros et al.) although the aim of the application role is to gather, in a single group, all the employees needing the same access rights to information, independently of their business role. As a result, the business role should always be perfectly aligned with the application role and the business tasks, which are performed by the employees, should always accurately correspond to the tasks defined in the business role.

With the second, the company has to continuously manage two types of employee assignments: the employee to business role assignment and that same employee to application role assignment.

This choice has been encountered in a health care establishment where each employee of a department is assigned to one of the business roles of the department and to a set of application roles defined at the IT level. The assignment to both types of role is realised at the recruitment of the employee, or after a modification of its business role. Due to the lack of formal alignment rules between the business roles and the application roles, it is frequent that when the assignment to a business task varies, the information is not automatically passed on to the IT department. This gradually generates access rights assignment errors.

From 1. and 2., we retain the following three problems:

- **insufficient analysis of the business roles,**
- **misalignment between the business roles and the application roles,**
- **misalignment between the employee's responsibilities and its access rights.**

1.5 Research questions and research objectives

The research aims at improving and completing the fields of business/IT alignment and access rights management by overcoming the lack of consideration of the needs of governance and of enterprise architecture reviewed in Sections 1.2 and 1.3. To reach these objectives, we consider the notion of **responsibility** as central to support the elaboration of the access rights and their deployments on the information system. Hence, this responsibility, which is motivated by governance frameworks and from human and organisational sciences, is used as a pivot between the business layer and the application one. Our perception of the responsibility is that it does not attempt to replace the role or to be a subset of it but rather, that it strengthens the link between an employee, its accountabilities related to a unique task, and its rights and permissions over the information system.

Thereby, the first research question that we address throughout this research is: **Considering the corporate and IT governance needs, what are the concepts which constitute the core of the responsibility and how these concepts may be associated in a dedicated Responsibility metamodel?**

The second research question that we address is: **How may business/IT alignment be improved considering the responsibility, in the context of enterprise architecture model, and for the field of access rights management?**

The latter research question brings about the following sub–question: **How may responsibility be mapped with the role based access control model and how does this mapping enhance the accuracy and the usability of the access rights management?**

By answering these questions, we aim at achieving the following three research objectives:

1.5.1 Definition of the employees' responsibilities

The review of the needs of governance argues for having the responsibilities of the employees defined along the enterprises' business layer (Section 3.2). The definition and the modelling of this concept of responsibility remains however insufficient and incomplete regarding to the many facets of governance. As a consequence, the first field impacted by the research concerns the definition of the employee's responsibilities. Our contribution, to exploit this concept, brings a new metamodel that includes and associates all the components of the responsibility. This Responsibility metamodel is built around the accountabilities of an employee regarding a single business task and around the rights and capabilities required to fulfil these accountabilities. Although, these concepts of business task and right are common in the field of IT, there is no explicit relation between them and the rights and capabilities provided to the employees are not systematically aligned with their accountabilities.

1.5.2 Enhancement of enterprise architecture models

The second field impacted is the field of the enterprise architecture models. As explained in Section 1.3, the advantages of the enterprise architecture models are manifold since they permit to better apprehend the structure of the company from top to bottom, including the interconnections between the business objects, the people, and the information system. However, the alignment between the layers of the enterprise architecture models is not always clearly explained, neither is it justified. To enhance these connections, our Responsibility metamodel is integrated in the enterprise architecture metamodel ArchiMate. The latter is an enterprise architecture metamodel used to give business and IT static views of the corporate architecture as well as the links between these views. An additional case study has also been performed related to the enterprise architecture model design at the European Court of Auditors. That integration allows enhancing the semantic richness of the concepts that compose the enterprise architecture frameworks.

1.5.3 Improvement of business/IT alignment

The third field impacted is the business/IT alignment that we consider in the sense of having the access rights on the information system accurately defined and assigned to the employees with respect to the business specifications. Section 2.2 and Section 2.3 highlight that there already exist models and methods that contribute to this field. However, most of them do not consider the governance needs because they are used at the application layer and because links with the business layer are infrequent. The fulfilment of the governance needs analysed in Section 3.2 during this activity is, consequently, insufficient.

In this thesis, we have decided to use the Responsibility metamodel to improve the interoperability between the business layer and the application layer of the ArchiMate enterprise architecture metamodel. Therefore, we consider the Responsibility metamodel for strengthening the links between both layers. At the business layer, responsibilities are defined according to business specifications, and at the application layer, access rights are managed based on these responsibilities. As a result, responsibility acts as an hyphen between the above mentioned layers.

1.6 Scope of the research and case studies

This section determines the type of companies targeted by the research as well as the resulting type of access rights they need. It also presents the two case studies which are used to illustrate and evaluate the research artefacts: the first case study concerns the Centre Hospitalier de Luxembourg (referred to as "the hospital") and the second concerns the European Court of Auditors (referred to as "the court").

1.6.1 Targeted companies

According to Mintzberg's framework, organisations may be differentiated along three basic dimensions (Lunenburg (2012)): the key parts of the organisation, its prime coordinating mechanism, and the type of decentralisation it employs. Based on these three dimensions, Mintzberg suggests five types of organisations: *simple structure*, *machine bureaucracy*, *professional bureaucracy*, *divisionalised form*, and *adhocracy*. *Simple structure* and *adhocracy* are very flexible and may have their functioning easily adapted according to the business evolution. *Divisionalised form* corresponds to organisations where the decision making process is at the division level and where the technostructure is located at corporate headquarters. *Professional bureaucracy* are relatively formalized and aims to provide high quality services. Employees from these organisations are highly qualified and the decentralisation is vertical or horizontal. These organisations correspond to hospitals or large law firms. *Machine bureaucracy* corresponds to organisations where the business is highly and formally defined by specific rules and procedures, and decisions are made following a vertical hierarchy. Typically, the organisations from this type (called "bureaucratic" in the thesis) correspond to big organisations such as large government administrations or steel companies, but also to smaller organisations like the logistic department in the hospitals or large urban school districts.

The bureaucratic organisations are necessary mostly when there exists, among others, a considerable need to carefully and formally manage, at the operational layer, a large amount of information. To protect this information, strict regulations, standards and norms (e.g., IT Governance Institute (2007), ISO27000 (2012) or Basel2 (2004)) have been elaborated and require to define the operational tasks, operational responsibilities and operational roles of the employees following the organisation processes. These standards and norms, which we are going to review later in the thesis, relate to different fields such as the management, the governance or the security of these organisations' operation layer.

The three research objectives explained in previous section are especially significant in this highly regulated environment and for these bureaucratic organisations. Therefore, our research is going to focus more specifically on how to enhance the business/IT alignment of these companies. For them, to have the access rights accurately defined and provided to the employees according to the operational responsibilities, and related to operational tasks, is a crucial requirement. Thereupon, the strategic or political responsibilities related to strategic activities (such as the one of the top manager of these institutions) are not directly in the scope of the research.

With regard to the type of alignment strategies depicted by Henderson and Venkatraman (1993), in Section 1.3, the business/IT alignment concerns the alignment between elements from the operational layers of the company rather than from the strategic layer. Hence, it focuses on:

the alignment between *the organisational infrastructure and processes* with *the IS infrastructure and processes*. This alignment aims among others, firstly to support the operational managers to accurately define the employee's responsibilities and the corresponding required access rights, and secondly to support the internal and external auditors by providing the motivations justifying the provided access rights.

1.6.2 Access rights "by design"

Given our focus on highly regulated companies, the access rights which we are going to address in this research concern the rights "by design" rather than right "on the fly". This means that we consider that in the companies where the access to the information is highly regulated, the access to the information is a right which is accurately and rigorously engineered following precise and well defined organisation artefacts such as the responsibility of the employees, their role, the task and the processes they have to achieve.

The same thinking led to consider that managing the rights "on the fly" (e.g., to face an exceptional situation) is a type of workaround used to provide access rights to the employee without complying with specific business rules and without any rigorous alignments with the business layer. Hence, this way of providing access rights is less conceivable in the frame of bureaucratic organisations and, as a result, is not going to be considered in the thesis.

1.6.3 Centre Hospitalier de Luxembourg

The first case study takes place at the Centre Hopitalier de Luxembourg[1]. The hospital is a public institute for serious pathological care, medical and surgical emergencies, and palliative care. The hospital also has an academic and a research orientation. In 2010, the hospital admitted 427,903 patients for consultations and outpatients, 25,532 inpatients, 33,277 adult and 31,857 paediatric emergency patients. On staff level, the hospital employs 2,046 staff including 152 specialist doctors, of which 55 have a liberal status, 53 cooperating doctors and 48 doctors in a specialisation process. The number of caregivers was 1336 and the number of administrative staff was 510.

In the hospital, having access to patients' records at the right moment is fundamental for the life of these patients. However, provisioning access rights to employees must be made under the constraints of confidentiality towards the patient data.

To face these constraints, the hospital has developed its own set of access control models based on the rule that the medical staff that accesses the patients' record must be associated to this patient, and if not the case, he must motivate the intervention that can justify the access. An innovative method for providing the access rights has been elaborated by the hospital. The method includes a data model structured on four confidentiality levels and scenarios to access patient's records adapted to each practitioner roles. In the first part of the case study, we have defined the responsibilities of each employee and we have analysed how these responsibilities allow providing access to the patient's records according to their needs.

[1]Translated into English by Hospital Centre of Luxembourg

In each hospital department, a job profile exists for the employees to specify the tasks they need to perform according to their business roles. In parallel, at the application layer, application roles are defined to manage the access rights, but their assignment to the employees is currently neither fully accurate nor justified. Therefore, the second part of the case study aims at analysing how this problem may be solved by defining accurate responsibilities and accountabilities to be assigned to each employees, considering the access rights they must be provided with, to perform these responsibilities and accountabilities.

1.6.4 European Court of Auditors

The second case study, which is presented in Appendix G, takes place at the European Court of Auditors, an independent audit institution of the European Union. The business role of the court is to carry out the audits of EU finances. In December 2011, the court employed 889 agents, 557 of them worked in audit chambers. The court uses its own enterprise architecture model to model the structure of its IS. The four layers of this enterprise architecture model are the process layer, the functional layer, the application layer and the data layer.

The IT of the court is structured in three units: User Services and Operations, Information Systems and Methods, and the IT Office. One role of the Information Systems and Methods unit is to provide the users with the access rights they need, based on their identity. As a European institution, the business roles at the court are strongly formalised and not directly connected to the real responsibilities and accountabilities of the employees. Therefore, at the origin, the access rights provided to the employees have been calculated based on the business tasks to be performed, without rigorous methods, and they have been progressively adjusted through time.

Since 2010, in order to support and enhance the access rights management activities, a project is on–going at the court which aims to automatically update the OIM tool (Oracle Identity Management) considering Sysper2 (application to manage the status of the employees) modifications. This update has called for a rework of the User Provisioning and User Account Management process.

During the case study, we have formalised the real responsibilities and accountabilities of the employees considering their assignments, and we have explained and demonstrated that it is much more consistent to provide access rights based on these well–defined responsibilities and accountabilities.

1.7 Research method

Improving the alignment of the business with the IT by defining the responsibilities of the employees using business information, and by aligning the access rights on the IS based on these responsibilities is a research that may plainly be considered in the scope of design science and action design research method.

Hevner et al. (2004) explains that the design science paradigm *seeks to extend the boundaries of human and organisation capability by creating new and innovative artefacts.* Four main artefacts are outputs from the thesis: a glossary and a Responsibility metamodel (Table 1.1), an integration of this Responsibility metamodel with the business layer of ArchiMate enterprise

architecture framework, and a method which exploits the integration of the responsibility in ArchiMate for the access rights management (Table 1.2).

Hevner maps his definition of artefact to the four research artefacts proposed by March and Smith (1995): construct, model, method and instantiation. This research framework proposed by March and Smith structures the research activities with a four by four table. This table permits to separate the research objects in sub–objectives and, hence, the research activities in sub–activities. Each sub–activity corresponds to a specific research section, which could be associated to a research method. The framework prescribes four research activities: build and evaluate, and theorise and justify, that may concern the four aforementioned outputs. The first two refer to design sciences activities, whereas, the latter two concern the natural science activities, out of the scope of this thesis. This approach, used to structure and evaluate de research, has already been used in many works (e.g.: Osterwalder (2004) and Edirisuriya (2009)).

	Build activity	**Build method**	**Built artefact**	**Evaluation method**
Construct	Analysis of the concepts which compose the Responsibility metamodel Ch.3.5	Library research in Computer Science, Library research in Human Science Ch.3.5	Glossary	Publications: Sec.2.4 and Sec.4.8
Model	Elaboration of the Responsibility metamodel Ch.4	Frameworks and Conceptual Models Ch.4	Responsibility metamodel	Interviews: Sec.5.5.5
				Publications: Sec.4.8

Table 1.1: Research methodology for main artefact 1 and 2

	Build activity	**Build method**	**Built artefact**	**Evaluation method**
Model	Integration of the Responsibility metamodel with the ArchiMate metamodel Ch.5	Conceptual mapping and integration Ch.5	ArchiMate extended with the Responsibility metamodel	Case study: Centre Hospitalier de Luxembourg Sec.5.5
				Interviews: Sec.5.5.5
				Publications: Sec.7.4
Method	Definition of an access rights management method based on the Responsibility metamodel Ch.6	Frameworks and Conceptual Models, Processes engineering Ch.6	Access rights management method based on the Responsibility metamodel	Case study: Centre Hospitalier de Luxembourg Sec.6.5
				Interviews: Sec.6.5.5
				Publications: Sec.6.6

Table 1.2: Research methodology for main artefacts 3 and 4

With respect to the framework, as shown in Tables 1.1 and 1.2, we focus on the build and evaluate research activities. In practice, we note that these activities are preceded, in most design science research methods, by an activity of problem discovering (e.g., Peffers et al. (2008)[1]). Concerning the elaboration of the Responsibility metamodel, given that this artefact originates from the organisation and must be closely linked to the other concepts from the business layer, we consider that the practitioners and end–users possess a rich knowledge regarding this field and that it is necessary to have them involved all along the artefact building activity. Therefore, we have considered the design research method proposed by Sein et al. (2011), named *Action Design Research* (Figure 1.5).

The action design research method has for objective to strengthen the connections between the practitioners and the researchers by combining the building, intervention and evaluation (BIE) activities. Accordingly, the method advocates for a continual evaluation of the problem and the built artefact in order to ceaseless adjust the artefact elaboration with real usage settings. In this thesis, given that the elaboration of the Responsibility metamodel has been informed by theories, we consider that we are in an *IT–Dominant BIE Generic Schema*[2] such as represented in Figure 1.5. In this schema, a first innovative artefact is created by the researcher and alpha versions are iteratively generated in a limited organisational context. In a second step, the more mature artefact is evaluated in a wider organisational setting and beta versions are shaped with the end–users.

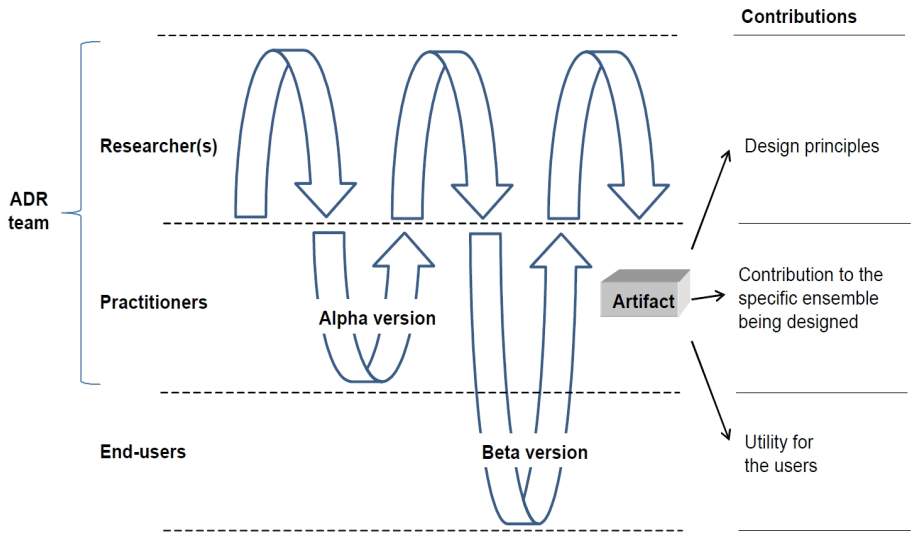


Figure 1.5: Action Design Research – The Generic Schema for IT–Dominant BIE, **Extracted from:** Sein et al. (2011)

[1] Peffers et al. (2008) call this activity: *Problem Identification and Motivation*

[2] The alternative to the *IT–Dominant BIE Generic Schema* proposed by Sein et al. (2011) is the *Organisation–Dominant BIE Generic Schema* which suits to design artefacts where *primary source of innovation is organisational intervention.*

1.8 Built artefacts

To be accurate, we have decided to divide the build column of the March and Smith research framework onto three columns, respectively, (1) the build activities that we perform to produce the artefact, (2) the build method that we use to realise the build activity and (3) the built artefact, which are the expected outputs of the research activities. The last column represents the evaluation method used to evaluate the artefact.

We exploit a precise type of research method for each research activity:

1. The analysis of the concepts of the Responsibility metamodel (Figure 4.13) is performed based on a double activity of library research:

   (a) A review of the existing literature about the responsibility in IS/IT sciences and in IS/IT frameworks is performed. This review of the concept of responsibility in IS/IT permits to understand how the responsibility is apprehended in computer science. In this field, the responsibility is often mentioned as a requirement but, its semantic and its deployment remain insufficient. The knowledge about responsibility from the field of IS/IT mainly concerns the obligations, and the rights and capabilities associated with the responsibility.

   (b) Then the library review is completed and improved, thanks to the analysis of the responsibility's concepts issued from social, managerial and psychological disciplines. We have decided to include the inputs from the human sciences since it brings a valuable knowledge contribution regarding some components of the responsibility, e.g., the accountability, the sanction and the answerability.

   The reason for choosing library research as a research method for the analysis of the concepts that compose the Responsibility metamodel is that, as defined by Palvia et al. (2003), it aims at summarising and synthesising past researches, and highlights some of the important conclusions.

2. The elaboration of the Responsibility metamodel aims at providing a set of concepts linked together in a coherent and justified way. Based on the analysis of the concepts that compose the responsibility, the appropriate concepts are integrated in the Responsibility metamodel. The selection is realised based on the way they are perceived in the literature. Afterwards, the coherent connections between the concepts are established and justified, again by confronting these connections with the different inputs from the literature. This research method corresponds to the method Frameworks and Conceptual Models proposed by Palvia et al. (2003) applied in the frame of action design research. For the elaboration of the Responsibility metamodel, this means that after having designed a first version of this metamodel, we have deployed it to one process at the European Court of Auditors. This deployment, explained in Appendix G, has allowed refining the Responsibility metamodel and its concepts (e.g., integration of new concepts such as the sanction, the actor, the governance rule and the source, and refinement of other concepts such as the task). Afterwards, following this enhancement, a beta version has been refined and evaluated at the Centre Hospitalier de Luxembourg, as illustrated in Chapter 6.

3. The integration of the Responsibility metamodel in an enterprise architecture model, ArchiMate, aims at facilitating the transition from the business layer to the application

layer. Therefore, the method used consists in conceptual mapping and integrations proposed by Parent and Spaccapietra (2000). In practice, to perform this integration, the correspondences between the concepts, and associations between the concepts, of both metamodels have been analysed, and the semantic and structural heterogeneities resolved.

4. The elaboration of the access rights management method, firstly, consists in defining an *Access rights management reference model*. This corresponds to the research method Frameworks and Conceptual Models proposed by Palvia et al. (2003) Secondly, according to this reference model, the elaboration of the access rights management method consists in engineering the processes necessary for the access rights management. In our case, this means understanding and formalising the processes performed by the access rights manager to assign access rights to the employees.

5. According to March and Smith (1995), the evaluation of a designed artefact must be realised based on precise evaluation criteria. Based on our research questions and objectives (Section 1.5), the criteria, and their definitions, which we have chosen are the following:

   - The Responsibility metamodel artefact has as objective to strengthen the representation of the responsibilities existing at the business layer and as a result, the modelling of the most significant information from this layer. Hence, the criterion associated to the evaluation of this artefact is the *expressiveness*. This expressiveness has been defined by Baker et al. (2000) as *the power to express complex information in ways that are easily understood*.

   - The integration of the Responsibility metamodel with the business layer of ArchiMate has as objective to support the exploitation of this metamodel. The criterion associated to the evaluation of this artefact is consequently the *usability*. This criterion has been defined in ISO9126-1 (2001) by the sub–characteristics of (1) understandability – *determines the ease of which the systems functions can be understood, relates to user mental models in human computer interaction methods*, (2) learnability – *learning effort for different users* and (3) operability – *ability of the software to be easily operated by a given user in a given environment*.

   - The method for the access rights management based on the Responsibility metamodel aims to enhance the exactitude of the engineering of the access rights provided based on the actors' responsibilities. Therefore, the criterion associated to the evaluation of this last artefact is the *accuracy*. In ISO9126-1 (2001), this criterion corresponds to the *accurateness* sub–characteristic which refers to *the correctness of the functions*, or the method in our case.

Concretely, these evaluations are performed with two case studies, as coined by Wieringa (2010), the first one at the Centre Hospitalier de Luxembourg and the second one at European Court of Auditors. Based on the case studies, the evaluations of the designed artefacts have also been performed by the practitioners from both institutions. These practitioner's evaluations have been realised by interviews conducted to the people that have been involved in the case studies at the hospital and at the court.

1.9 Structure of the thesis

This thesis is structured in three parts: Part I presents the state of the art and analyses the governance needs, Part II presents the elaboration of the Responsibility metamodel and the integration with the business layer of ArchiMate, and Part III presents firstly an alignment of the Responsibility metamodel with the RBAC model and, secondly, a method for the access rights management based on this alignment and considering the requirements from the business layer. Additionally, Chapter 1 introduces the context and the research method and Chapter 7 evaluates and concludes the research and provides ideas for future works. The Figure 1.6 provides a view of the chapters, as well as how they contribute to each other.

1.9.1 Part I

The heart of this research aims to improve the definition of the access rights assigned to the employees based on the new governance needs. Part I includes, as a consequence: the state of the art in the existing models and methods for the management of the access rights, the review of the needs of governance, and a review of the fundamentals of responsibility.

- In Chapter 2, we make a state of the art in two fields: the field of access control models (**Item 4** of Figure 1.4) and the field of rights engineering methods (**Item 3**).

- In Chapter 3, we analyse some arising professional governance frameworks for the management of the enterprise (**Item 2**) and for the information system (**Item 1**). This analysis provides us with a list of governance needs that we need to deal with for the access rights management. These needs concern directly the access rights (**Item 3 and 4**) or more global governance's aspects. In this chapter, we review the concept of responsibility and the concept that composes the responsibility through the different disciplines to elaborate the Responsibility metamodel in Chapter 4.

1.9.2 Part II

Part II of the research encloses the elaboration of the Responsibility metamodel and the integration of it with the business layer of ArchiMate.

- Chapter 4 presents the Responsibility metamodel that we have elaborated and which we have named ReMMo. The chapter introduces the metamodel in UML and proposes a glossary of the concepts integrated in the metamodel. We also illustrate the instantiability of the Responsibility metamodel with regards to the healthcare domain.

- In Chapter 5 we propose an integration of the Responsibility metamodel with the business layer of ArchiMate to allow defining employees and business roles responsibilities and to enrich the semantic of the connections between the concepts of ArchiMate.

  We evaluate the expressiveness of the Responsibility metamodel and illustrate how the integration with ArchiMate may be instantiated with the first part of the case study at the Centre Hospitalier du Luxembourg. This case study concerns the definition of the responsibilities of the employees from the hospital.

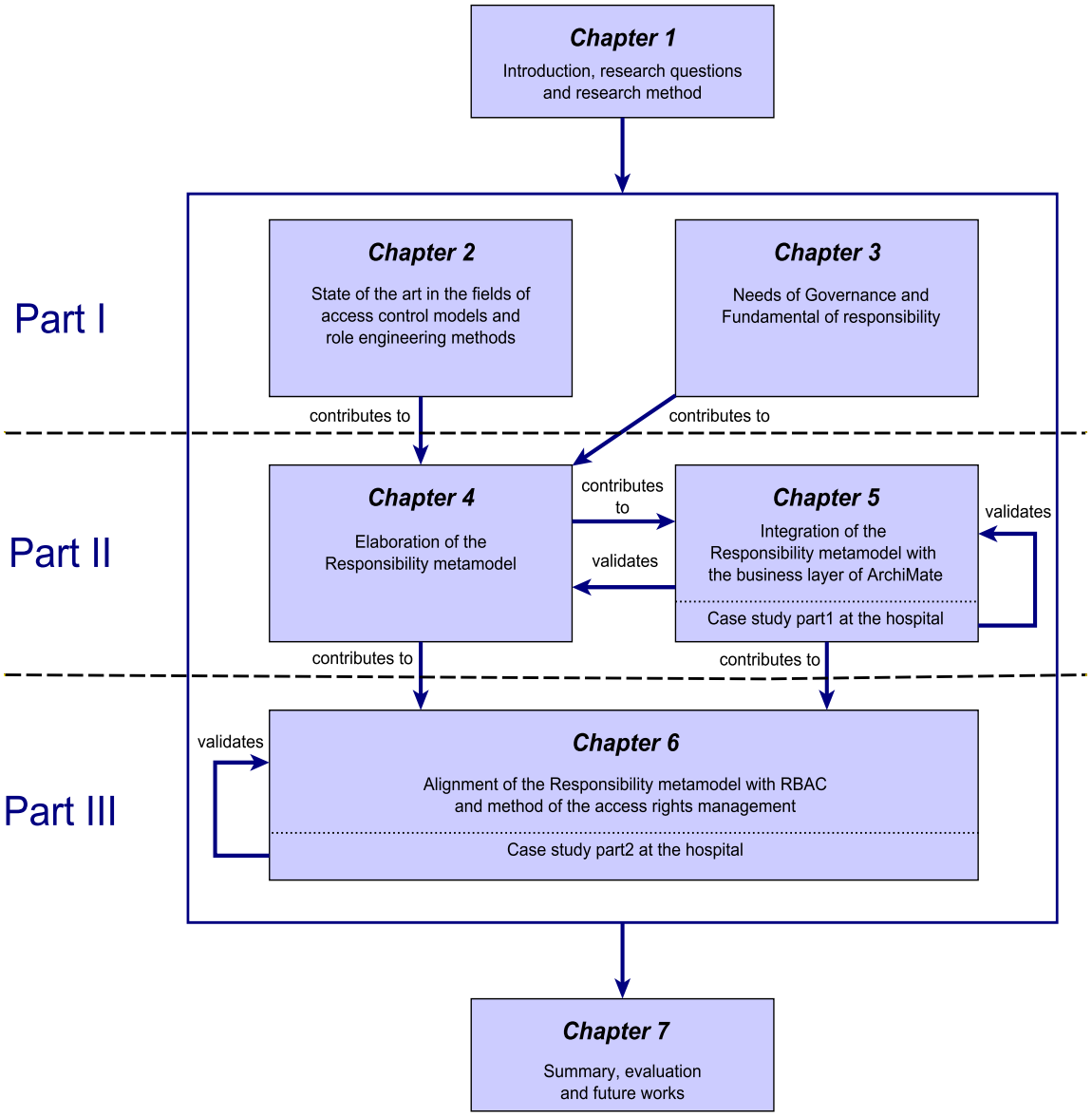


Figure 1.6: Structure of the chapters in the thesis

1.9.3 Part III

Part III of the research concerns the definition of an access rights management method based on the Responsibility metamodel.

- In Chapter 6, firstly, we propose a method for the access rights management. Therefore, we depict the *RBAC reference model* proposed by Band (2011). Afterwards, we align the Responsibility metamodel with the RBAC model to analyse which concepts from the first metamodel generate concepts from the second. Then, we propose a set of processes to instantiate the *RBAC reference model* considering this alignment.

  The second part of the case study at the Centre Hospitalier du Luxembourg evaluate that the integrated ArchiMate with the Responsibility metamodel enhances the definition of the access rights and that the definition of the responsibilities may be used to generate RBAC roles and permissions.

Chapter 7 summarizes the research results, evaluates the research activities through the seven Hevner's guidelines and provides ideas for futures work.

1.10 Publications

Publications related to this thesis:

- C. Feltus, Preliminary Literature Review of Policy Engineering Methods – Toward Responsibility Concept, in *Proceedings of the International Conference on Information and Communication Technologies: from Theory to Applications (ICTTA)*, Damascus, Syria, 2008. IEEE.
- C. Feltus, M. Petit, Building a Responsibility Model Including Accountability, Capability and Commitment, in *Proceedings of the 4th International Conference on Availability, Reliability and Security (ARES)*, Fukuoka, Japan. 2009. IEEE.
- C. Feltus, M. Petit, F. Vernadat, Enhancement of CIMOSA with Responsibility Concept to Conform to Principles of Corporate Governance of IT, in *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, Moscow, Russia. 2009.
- C. Feltus, M. Petit, E. Dubois, Strengthening Employee's Responsibility to Enhance Governance of IT – CobiT RACI Chart Case Study, in *Proceedings of the 1st Workshop on Information Security Governance (WISG CCS)*, Chicago, Illinois, USA. 2009. ACM.
- C. Feltus, M. Petit, M. Sloman, Enhancement of Business IT Alignment by Including Responsibility Components in RBAC, in *Proceedings of the 5th International Workshop on Business/IT Alignment and Interoperability (BUSITAL)*, Hammamet, Tunisia. 2010.
- C. Feltus, E. Dubois, M. Petit, Conceptualizing a Responsibility based Approach for Elaborating and Verifying RBAC Policies Conforming with CobiT Framework Requirements, in *Proceedings of the 3th International Workshop on Requirements Engineering and Law (RELAW10)*, Sydney, Australia. 2010. IEEE.
- C. Feltus, M. Petit, E. Dubois, *ReMoLa*: Responsibility Model Language to Align Access Rights with Business Process Requirements, in *Proceedings of the 5th International Conference on Research Challenges in Information Science (RCIS)*, Guadeloupe – French West Indies, France. 2011. IEEE.

- C. Bonhomme, C. Feltus, M. Petit, Dynamic Responsibilities Assignment in Critical Electronic Institutions – A Context–Aware Solution for in Crisis Access Right Management, in *Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES)*, Vienna, Austria. 2011. IEEE.
- C. Feltus, A. Khadraoui, A. Yurchyshyna, M. Léonard, E. Dubois, *Responsibility aspects in service engineering for eGovernment*, in *Proceedings of the Workshop of the 6th Interoperability for Enterprise Systems and Applications conference (I–ESA), Service Science and the next wave in Enterprise Interoperability*, Valencia, Spain. 2012.
- M. Petit, C. Feltus, F. Vernadat, Enterprise Architecture Enhanced with Responsibility to Manage Access Rights – Case Study in an EU Institution, in *Proceedings of The Practice of Enterprise Modeling – 5th IFIP WG 8.1 Working Conference (PoEM)*, Rostock, Germany. 2012.
- C. Feltus, E. Dubois, E. Proper, I. Band, M. Petit, Enhancing the ArchiMate® Standard with a Responsibility Modeling Language for Access Rights Management, in *Proceedings of the 5th International Conference on Security of Information and Networks (SIN)*, Jaipur, Rajastan, India. 2012. ACM.

Chapter 2

State of the art in access rights models and rights engineering methods

2.1 Introduction

The objective of this chapter is to review the state of the art in the field of access control models (**Item 7** of Figure 1.4) and in the field of the business/IT alignment related to the rights engineering methods (**Item 3**). This review is achieved to motivate our research and to figure out the level of integration of these models and methods with the organisational artefacts (**Item 6**) and, hence, their integration with the arising governance needs (**Items 1 and 2**) reviewed in Chapter 3 that is afterwards analysed in Section 3.4.

To analyse the state of the art, we have reviewed the existing state of the art achieved in the field of access control models and in the field of rights engineering methods. We have also reviewed the main sources of knowledge addressing this field, including a list of the most significant conferences concerned. This list of conferences includes: SACMAT[1], the Policy Workshop[2], ARES[3], CCS[4] for the domain of the access control models and CAiSE[5], RE[6] or BusITAl workshop[7] for the domain of the business/IT alignment. These sources of knowledge have been scanned to retrieve the most appropriate papers which could contribute to our state of the art.

This chapter is structured as follows: the next section introduces an analysis of the most significant access control models, to understand their particularities and functional utility. The review has been sized to the most acknowledged models that we have selected, based on O'Connor and Loomi (2011). The analysis highlights the concepts used for the access rights management

[1] http://www.sacmat.org/
[2] http://www.policy-workshop.org/
[3] http://www.ares-conference.eu/
[4] http://www.sigsac.org/ccs.html
[5] Edition of 2012: http://www.caise2012.univ.gda.pl/
[6] http://www.requirements-engineering.org/
[7] Edition of 2011: http://siti-server01.siti.disco.unimib.it/busital2011/

through the different models, as well as the links between them. Afterwards, Section 2.3 makes a review of the business/IT alignment methods centred on the access rights definition and on the access rights assignment to the employees. Most of the reviewed methods consider the access rights based on the concept of the role of the employees and thus as a consequence, are also termed "roles engineering methods".

2.2 Access control models

Many states of the art have already been proposed in the field of access control models (Arribas (2003), Crook et al. (2003) and Fuchs et al. (2011)). The first developments in this field have been performed by Lampson (1974) who has introduced actual access control basic principles and, operating systems and distributed system controls. A collection of models has been defined subsequently. The models that we analyse are the Mandatory Access Control (MAC), the Discretionary Access Control (DAC), the Role Based Access Control (RBAC), the Usage Control model (UCON) and the Attribute Based Access Control (ABAC) model, models based on the concept of task, temporal dependencies, organisation, and team.

2.2.1 Mandatory Access Control

In mandatory access control, only one authority manages the access rights and the users are not allowed to modify them (Arribas (2003)). MAC is also called Ruled–based Access Control and it defines the concepts of objects and subjects which are classified in classes and levels. Objects are resources to protect (passive) and subjects are active entities which access these objects. Subjects are processes or programs which are activated by the users. MAC policies are also known as *Multilevel security policies*. The subject and the object classification normally refers to *security classification* or *information flow policies*. MAC has been particularly used in the military domain where hierarchy is fixed.

One special MAC policy is the Lattice–based access control. In this control model, objects and subjects are classified into access classes and each access class is associated with a security level and a set of categories. The security level determines the sensibility level of those objects and subjects, e.g., *TopSecret*, *Secret*, *Confidential* and *Unclassified*. The set of categories corresponds to domains of competence or to functions, e.g., *Army*, *Navy*, *Nuclear* and *Administration*. *Army* includes *Navy* which includes *Nuclear* which includes *Administration*. There exists a dominance relation between access classes. This relation is written as $\geq$. The access class c1 dominates access class c2 if, and only if, the security level of c1 is higher or equal to the security level of c2 and if this category c1 includes the category c2.

The Bell–LaPadula model (Bell and La Padula (1976)) finds his foundation in MAC model and is strongly linked to the MAC's concepts (Figure 2.1). The system is composed of subjects, objects and actions. Each object is associated with an access class that defines its sensitivity level and each subject is associated to an access class called *Clearance*. The actions are performed by the subject on the object, e.g., read–write (Davrondhon Gafurov and Svendsen (2005)). The principle of the model is: no–read–up (a subject may only receive read access on an object if the subject access class dominates the object access class) and no–write–down (a subject may only receive write access on an object if the object access class dominates the subject access class). This means that the information flow may only be achieved from the lower classes up to

the higher classes and thus, by consequence, it enforces the confidentiality. Mclean (1987) has highlighted in 1987 that the Bell–LaPadula model is not completely secure because of a possible modification of the state of one object or one subject during a transaction. The Bell–LaPadula model provides the access control rules based on access classes which are created and enforced by one trusted user. The model may also be associated, additionally, to the discretionary access control model to enhance the flexibility of the rights provided (Fan et al. (2009)).

Mandatory access control models are currently implemented in products such as: SUSE Linux, UBUNTU, Window Vista, FreeBSD or SELinux (Security–Enhanced Linux) which has added a mandatory access control architecture to the kernel of Linux .

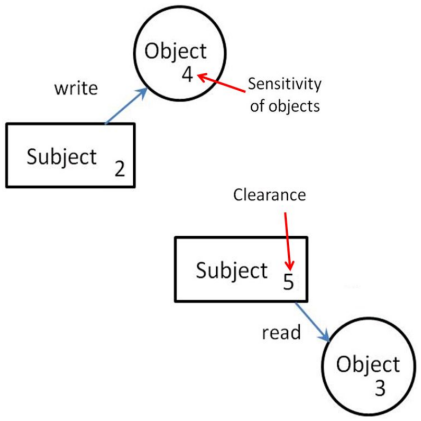


Figure 2.1: Bell–LaPadula model

While the Bell–LaPadula model ensures confidentiality, the Biba model proposes an additional control that ensures the integrity (Biba (1977)). The Biba principle is to assign each object or subject to an integrity class, e.g., crucial, important, or unknown. Biba considers that, at the user level, the integrity class reflects the trust level associated to the user to introduce, modify, or delete an object. It also considers that at the object level, the integrity class reflects the trust level associated data included in the object, and consequently, the degree of damage caused by an unauthorised modification of this object. According to the Bell–LaPadula model, Biba's model may be defined as: No–read–down, no–write–up.

In 1982, Lipner (1982) also tried to develop a model to ensure integrity based on the Bell–LaPadula model. In this model, Lipner illustrates how the Bell–LaPadula model and the Biba model can be combined, to develop a model usable for commercial applications.

2.2.2 Discretionary Access Control

A Discretionary Access Control model defines the concept of subject, object and action. Unlike MAC model where access control is managed by a central authority, in the discretionary access control model (DoD (1985)), a subject can receive access control rights and, as a consequence, can define access control on well defined objects. In principle, the subjects are identified and rules exist to specify which subject may perform which action on which object. This control is also called Identity–Based Access Control (IBAC).

One declination of DAC is the access matrix (Covington et al. (2001)). The access matrix defines subjects with privileges on objects (Figure 2.2). By definition (Hu and Scarfone (2012)), the privilege aims at reducing *the access space from a space where any authenticated subject can access all information to a space where specific users can only perform specific actions on specific objects*, e.g., *Subject V* may *read–write Object G*.

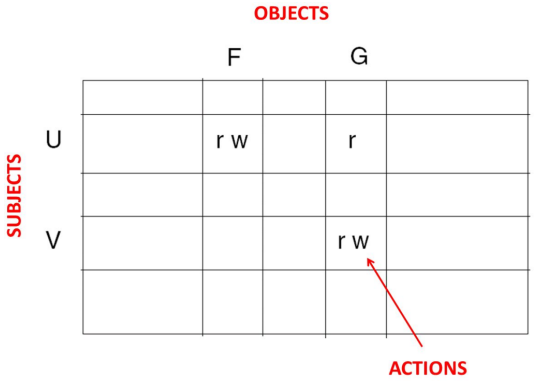


Figure 2.2: Access matrix

The HRU model (Harrison et al. (1976)) is a particular access matrix formalisation (HRU is an acronym of Harrison, Ruzzo, and Ullman). The HRU model defines six primitive operations: *enter* or *delete* an action, *create* or *delete* an object or a subject. The access matrix can become difficult to manage and can, in some cases, contain a huge amount of free cells, which makes it inefficient. Three approaches propose solutions to tackle this problem: the Authorisation Tables, the Access control lists (ACL), and the Capabilities.

The Authorisation tables are composed of tuples (user, privilege, and object). Each entry (or tuple) corresponds to a privilege that a user possesses on an object. Authorisation tables generally are used for the DBMS (Data Base Management System). They permit to reduce the size of the matrix. With the ACL approach, an object (e.g., a file) owns a list of each subject's privileges regarding this object, e.g., ACL is implemented in Unix and is formulated by the syntax *rwx r-x rw-* (respectively for *user*, *group*, and the *rest of the world*). ACL is also supported by the version 10.4 of Mac OS X. In case of capabilities, a list is associated to each subject. E.g., the user X has access to file A in read mode, to file B in read, write and "owner" mode, and so forth. Access rights management based on Capabilities has been exploited, amongst others, by the IBM AS/400 technology.

In 2003, the DAC model has been subject to improvements:

- Addition of conditional authorisations: authorisations can be system–dependent (e.g., place where the user accesses the system), content–dependent (e.g., in a database, the access can be restricted to some objects), history–dependent (e.g., access is provided based on the content previously consulted) and temporary authorisation. Moffett (1994) motivates the need to model and represent the (DAC) policies and their attributes such as the policy constraint of the temporary authorisation. Moffett proposes to describe the policies as objects and thereby, allowing them to be created, modified and queried.

- The concept of user and object group: authorisations are granted to groups of users or to groups of objects and the users inherit the right(s) of the group(s) they are assigned to.
- The concept of positive and negative authorisations. In the case of closed policy, the access is granted if a positive authorisation exists otherwise it is denied and in the case of Open policy, the access is granted if a negative authorisation (forbidding) does not exist. A mix of positive and negative authorisations can be used (e.g., a group is authorised to access a resource, but some members of the group are not). Using the positive and negative authorisation can, however, be a source of inconsistencies when a user has, at the same time, a positive and negative authorisation.

One important characteristic of the DAC model is the way it is managed. Many types of management exist: centralised, hierarchical, cooperative, ownership, decentralised, and so forth. Decentralisation is an interesting approach because it introduces the concept of delegation.

Models based on specific requirements, such as the integrity, have also been developed for the DAC model and are similar to Biba's model for the MAC model. The more renown is the one from Clark and Wilson in 1987. The Clark and Wilson's model defines a set of rules based on the practices related to commercial database processing which has the objective to keep the customer's data integrity (Clark and Wilson (1987) and Ge et al. (2004)).

The Chinese wall security policy (Brewer and Nash (1989b)) is aligned with some rules defined by Clark and Wilson (1987). The objective of this model is to assure that when two entities (e.g., Company Y and company Z) are in the same *conflict class* (the set of all companies whose corporations are in competition), and when an object A from entity Y is accessible to a user, object B from entity Z would no longer be accessible by the same user since there exists a conflict of interest between both companies.

2.2.3 Role Based Access Control

The objective of the Roles Based Access Control (RBAC) model is to create policies consistent with the organisational structure of the company. RBAC assumes that the most frequent way to access a resource is not the user's identity but the role that the user is assigned to in the company (Ferraiolo et al. (2001)). A role is defined as follows: *a role is a job function within the context of an organisation with some associated semantics regarding the authority and the responsibility conferred on the user assigned to the role.* This means that a user is associated to a role and that the permissions are also associated to the role.

The first works relates to the idea of grouping privileges to access resources have been performed by Baldwin (1990) who introduced the Names protection domains (NPD). These NPD have, afterwards, evolved toward roles.

The RBAC model is a junction of many models: core RBAC (RBAC0), hierarchical RBAC (RBAC1), constrained RBAC (RBAC2) (static separation of duty relations and dynamic separation of duty relations) and constrained RBAC with role hierarchies (RBAC3).

The core RBAC model (Figure 2.3, extracted from Ferraiolo et al. (2001)) is composed of the basic elements which define the model: the concept of users (USERS), representing mainly hu-

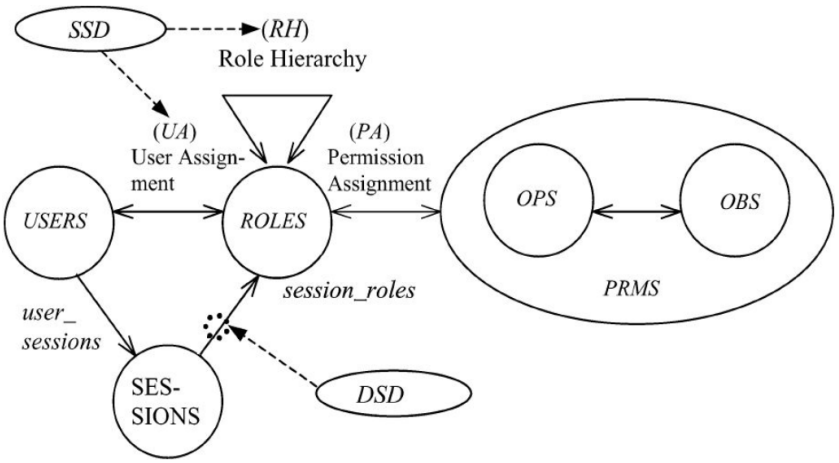


Figure 2.3: Role Based Access Control model, **Adapted from:** Ferraiolo et al. (2001)

mans, the concept of roles (ROLES) (see definition above), the concept of objects (OBS) which corresponds to accessed resources, the concept operations (OPS) which are computer–system functions executed in the name of the users, the concept permissions (PRMS) which corresponds to access permissions over the objects of the system and which, as a consequent, defines the links between the objects and operations.

The core RBAC also defines two types of relations between those elements. The first one is the user assignment (UA) which defines the relation between user and role. One user can be assigned to several roles and one role may includes many users. The second relation is the permission assignment (PA) which defines the relation between role and permission. Here too, one role may be assigned to many permissions and one permission may be provided to many roles.

Additionally, a connection amongst a role and a user is realised through a session (SESSION). The principle is that when a user logs onto a system, he must activate a number of roles to get the permissions associated to these roles. Sessions are, by consequence, means for the users to log on in a minimal time to perform the tasks. This mapping is achieved with two functions: the *user_sessions* which connects a session and the unique users of the session, and the *session_roles* which connects a session and a role active in the session.

The roles hierarchy model (Figure 2.3, extracted from Ferraiolo et al. (2001)) enhances the core RBAC model by a hierarchy between the roles. This model is a significant aspect of the RBAC model because it reflects the organisation and the hierarchical model of this organisation. It permits to define the relations between the roles and consequently, the possibility for the roles to inherit privileges of other roles. A hierarchical relation between the role r1 and the role r2 such as r1 is hierarchically higher than r2 makes r1 inherits all r2 privileges.

Ferraiolo, Sandhu and Gravila proposed in 2001 (Ferraiolo et al. (2001)) two kinds of role hierarchy: the General role hierarchy and the Limited role hierarchy. The General role hierarchy's main idea is that role hierarchy can be an arbitrary partial order and that multiple inheritances are allowed. In this case, a user may receive permissions from many roles, e.g., a

secretary inherits the permissions of the administrative staff role and of the employee role. In some cases, a user may inherit conflicting roles (roles that may not be activated at the same time for specific reasons). As a consequence, the notion of Limited role hierarchy has been introduced. The Limited role hierarchy's main idea is that the hierarchy can only be a tree structure and multiple inheritances are disallowed.

The last model composing of RBAC is the constrained RBAC model (Figure 2.3, extracted from Ferraiolo et al. (2001)). This model is based on the principle of the separation of duties introduced by Clark and Wilson (1987) which has for objective to avoid, for some users, to get permissions associated to conflicting roles. The separation of duties is declined in static and dynamic separations of duties. The static separation of duties model includes the definition of a set of roles and the constraint that if a user is assigned to a role of this set, he cannot be assigned to another role of this set at the same time. More precisely, the model may consider two arguments: a *set of minimum two roles*, and *a cardinality*. The latter limits the number of roles that can be assigned to a user in the same set of roles.

In the dynamic separation of duties model (Figure 2.3, extracted from Ferraiolo et al. (2001)), the constraints related to the assignment of roles to a user is a function of the role(s) that is/are already assigned to the user during the same session. In the static separation of duties model, the assignment constraints of user to role is existing independently of the session.

Many solutions have been proposed to represent RBAC and to support reasoning about it. Zhao et al. (2005) propose a logical approach for representing the model, for verifying correctness of the policies specified and for making access control decisions. In Massacci (1997), the author defines, on the one hand, a logic including a language and a semantic to *express RBAC policies in a simple and natural way*. On the other hand, he proposes a decision method to verify the consistency of these policies. Jajodia et al. (2001) propose a framework to enforce multiple access control policies using a specific language. Based on this language, users can specify security policies to be enforced according to different strategies and related to the security policy needs and the different users, groups, objects, or roles to which it applies. Ahn and Sandhu (2000) have defined the Role-based Constraints Language (RCL 2000). This latter allows formalising RBAC constraints in an intuitive manner and translating them into a first order logic restricted form expression.

Practically, RBAC has been implemented in many products, e.g., FreeBSD, Solaris, Microsoft Active Directory and Microsoft SQL Server, SELinux, PostgreSQL 8.1, SAP R/3, or Oracle DBMS.

2.2.4 Attribute Based Access Control

Although many researchers agree upon the advantages of RBAC, the literature also provides some disadvantages of it like in Covington and Sastry (2006), Lang et al. (2008) and Kuhn et al. (2010) that summarise the difficulties encountered by RBAC in order *to set up an initial role structure*, on the first hand, and for inflexibility in rapidly changing organisational domains, on the other hand. In addition, the RBAC model does not provide an appropriate solution for the management of dynamic business attributes such as the user credentials, e.g., space location, accessing time, a clearance level or a citizenship (Karp et al. (2009)).

ABAC (Attribute Based Access Control) is a solution that has been developed in order to meet this dimension. The two significant components of the ABAC model (Figure 2.4) are the attributes and the rules used to calculate (based on the attributes) the access decision.

The concepts modelled in ABAC are the concepts of object, subject, object attribute, subject attribute and permission (Jin et al. (2012)). The subjects are created by users and perform actions on the system. The subject attributes are values that are constrained by the system and/or are inherited from the user's attributes (the resource that uses the system). These attributes are for instance the name, the role, the user ID, the clearance, and so forth. The object is the passive entity which needs to be protected and which possesses object attributes such as an access condition. Permissions are privileges that the user may hold on an object. Based on the above explanation of the concepts, we deduce that objects and subjects both exist at the application layer although the user which uses the system is a concept that has an existence at the business layer.

The main advantage of ABAC is that role engineering is no longer necessary since the access decision is performed based on the user and the context information. The disadvantage is the dynamic of the access control policies that makes it difficult to audit the system (O'Connor and Loomi (2011)) or even to make risk analysis (Kuhn et al. (2010)).

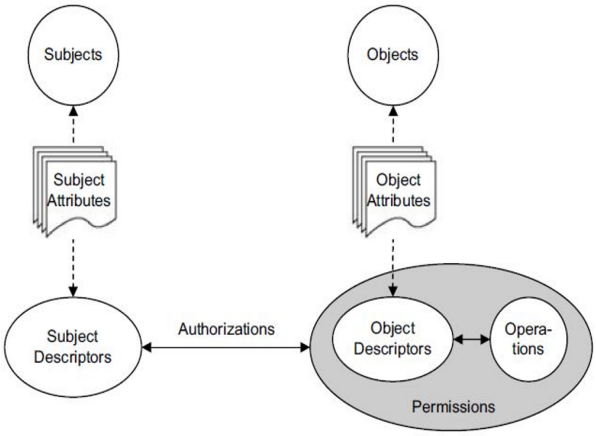


Figure 2.4: Attribute Based Access Control model, **Source:** Priebe et al. (2007)

ABAC has been tested in different areas where the traditional access control models are not sufficient to include all the constraints of the domain. Lang et al. (2008) has for instance applied it in the field of grid computing and has considered the multiplicity of the policies. Covington introduces a model for achieving attribute–based authorisation considering situation awareness and contextual attributes in the frame of mobile computing in Covington and Sastry (2006). As ABAC is motivated by its flexibility and ability to be used through many platforms and applications, the necessity to have a common understanding of the attributes is necessary. Many researchers have also addressed this challenge (Priebe et al. (2006)).

XACML[1] is the most appropriate standard to implement ABAC.

2.2.5 Supplementary access control models

Additional access control models have been defined with the aim to achieve specific purposes. We review four of them: TBAC, TRBAC, OrBAC and TMAC.

2.2.5.1 Task–Role Based Access Control

The access control has also been investigated through the medium of the Task component. Thomas and Sandhu (1998) introduce TBAC, a Task–based Authorisation Controls dedicated for "active security model". These models are opposite to the traditional model such as RBAC in that permissions may be granted along the activation of business activities and tasks. In TBAC, access control is calculated based on contextual information which is embodied in the authorisation–steps on the first hand, and on the usage and validity counts on the other hand. The authorisation–step is the main step of TBAC. It aims at grouping trustees and permissions. The usage allows the activation or deactivation of permissions according to the tasks which are allowed to be performed, in function of the contexts such as a work–flow, dependencies or task instances.

Task–Role Based Access Control (Oh and Park (2003)) (TRBAC) is motivated by new advancements in technology and services, and by the necessity to automate the supply of appropriate rights for tasks and services distributed on the network. TRBAC aims at assigning permissions to roles in an enterprise environment by considering the concept of task that is perceived as the fundamental unit of a business process. TRBAC associates permissions to tasks and to groups of users with the same role which operate the same tasks. Rather than directly accessing the business objects, such as with RBAC, the users perform the business process through business tasks to which permissions are associated. Four types of assignments are defined in the model, the User–Role Assignment (URA), the Task–Role Assignment (TRA), the Task–Work–flow Assignment (TWA) and the Permission–Task Assignment (PTA). Oh and Park (2003) consider that the TRBAC concepts of user, role and task correspond to the concepts of user, business role and business task from the business layer. The concept of permission is defined as *a read or write privilege for a file*. Hence, this permission and the object on which it applies exist at the application layer.

Gaaloul and Charoy (2009) propose an alternative Task–oriented Access Control models which is based on RBAC and which allows granting authorisation based on work–flow specifications and user authorisation information.

2.2.5.2 Temporal–Role Based Access Control

Temporal–Role Based Access Control (Bertino et al. (2001)) (TRBAC) has the same acronym as the Task–Role Based Access Control but it is an extension of RBAC which addresses the dynamic aspect existing in the assignment of permission to role. Bertino et al. explain that TRBAC aims at allowing enabling and disabling periodic role and *temporal dependencies among*

[1] eXtensible Access Control Makeup Language, http://xml.coverpages.org/xacml.html

actions, expressed by means of role triggers. TRBAC considers that the user and the temporal information are directly derived from the business layer and are represented at the application layer. Objects may be physical or logical and thus may exist at the business or application layer. However, permissions are calculated at the application layer. A Spatio–Temporal Role Based Access Control Model has also been proposed by Ray and Toahchoodee (2007) to address security requirements of pervasive computing applications. The authors associate each component of RBAC with spatio–temporal information and formalise the model by enumerating constraints.

In the thesis of James B. D. Joshi (2003), a Generalised Temporal Role Based Access Control (GTRBAC) model is proposed to ally RBAC model with a powerful temporal framework.

2.2.5.3 Organisation Based Access Control

Organisation Based Access Control (OrBAC) is an access control model developed with the objective to allow the definition of security (and access rights) policies independently of the application layer (Cuppens and Miège (2003)), and, therefore, OrBAC defines two levels. A concrete level which intuitively corresponds to the application layer and which is composed of the subject, the action, and the object and an abstract level which intuitively corresponds to the business layer and which is composed of the role, the activity, and the view. The relation between those concepts is such that the role abstracts the subject and the subject is empowered to role, the activity abstracts the action and the action is considered in the activity, and finally, the view abstracts the object and the object is used in the view. At the abstract level, abstract security policies are defined independently of the organisation. At the concrete level, the policies are specified in concrete policies according to this organisation. Thereby, the security policies may be defined in modular ways.

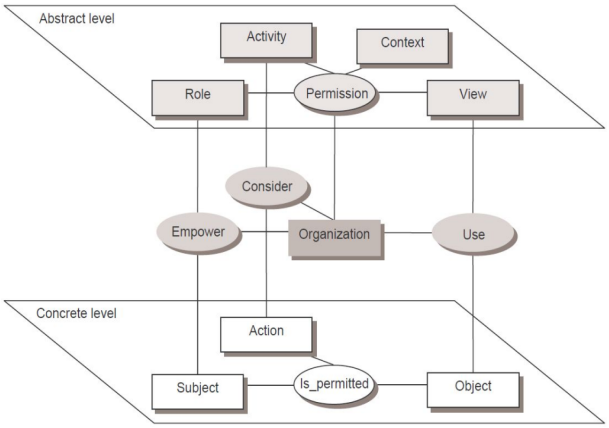


Figure 2.5: Organisation Based Access Control, **Source:** http://en.wikipedia.org/

These relations are moreover engineered in a context which is existing into an abstract level. This context allows adapting the policies according to specific circumstances like the date or the space, or according to functional circumstances like, e.g., an information system which works in a degraded mode following a security policy. As OrBAC is context sensitive, the policy could be expressed dynamically. Figure 2.6 highlights an ontology of context supported by OrBAC.

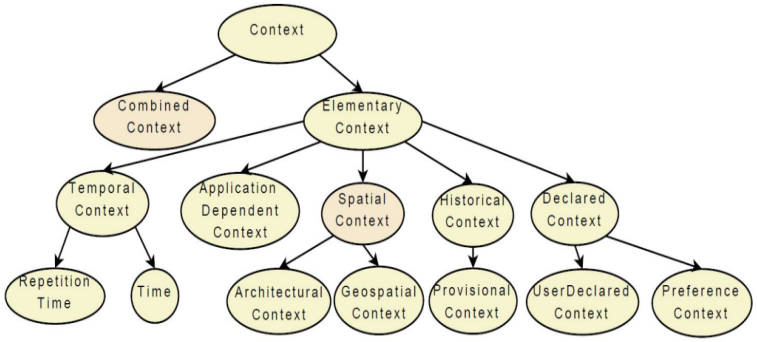


Figure 2.6: Context ontology and components, **Source:** Cuppens et al. (2007)

OrBAC allows modelling concrete and abstract security policies and allows the policies to be of the types authorisation or prohibition. Thereby, conflicts may happen between policies from the abstract or concrete level. However, Cuppens et al. (2007) explains that if conflicts are solved at the abstract level, they may not appear at the concrete one.

MotOrBAC is a tool that has been developed in order to design, analyse, implement, simulate and administrate security policies based on the OrBAC model. This tool has been presented in Autrel et al. (2008).

2.2.5.4 Team–Based Access Control

TeaM–based Access Control (TMAC) has been defined for the management of the access rights in collaborative environments and is based on the concept of a team which gathers a set of users in specific roles and in the context of a well defined task to do, or goal to achieve Thomas (1997). TMAC benefits of the advantages of RBAC while allowing refining the permissions assignment to objects in functions of tasks defined in a specific context. This "collaboration" context is twofold. Firstly, it applies on the user (the user context), e.g., user is member of a team, and secondly, it applies on the object (the object context), e.g., the set of object instances necessary by a team to perform a task. The advantages of TMAC are the scalability of the security administration and the possibility to assign permissions to users based on objects instances. The concepts exploited in the model are the team and the user which represent elements from the business layer but which exist at the application layer. The concept of object and permission exist at the application layer.

TMAC has also been subject to extensions, in Georgiadis et al. (2001), the authors propose a framework termed C–TMAC to integrate TMAC with RBAC, and to consider additional general context information such as the time or the location (of the object or of the subject).

2.2.6 Usage Control

UCON model (Usage Control model) was introduced in 2002 by Park and Sandhu (Park and Sandhu (2002)). The term *Usage* means usage of rights on digital objects. UCON gathers in a single model many: traditional access control models, the trust management and the digital

rights management (DRM). The traditional access control (DAC, MAC, RBAC, and so forth) represents the control in a closed system where users are identified. This control is realised at the server side. The trust management is used for the assignment of authorisations to subjects out of the system in the frame of an open environment such as the Internet. The DRM assures the access control to digital information and the control is realised on the client side.

These three models are complementary and aim at achieving different targets. Park and Sandhu make the statement that needs evolve and that consequently, some cases need the use of the three models simultaneously, which justifies the UCON model (Figure 2.7).

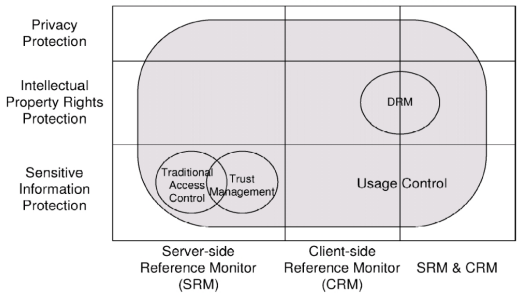


Figure 2.7: Coverage of UCON, **Source:** Park and Sandhu (2004)

Many representations of the UCON model components exist (Figure 2.8). Park and Sandhu have presented the first in SACMAT'02 (Park and Sandhu (2002)). Afterwards, a reviewed version that integrates the subjects–attributes and the objects–attributes (Sandhu and Park (2003)) in 2003 is illustrated in Figure 2.9.

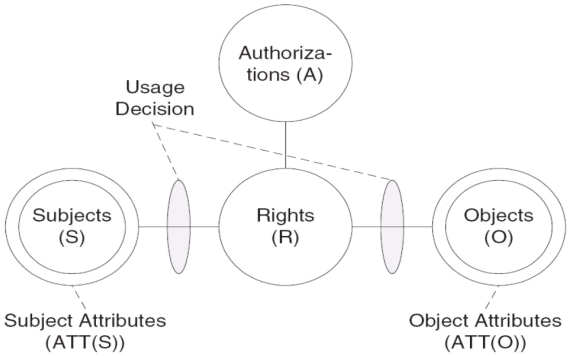


Figure 2.8: Traditional access control model, **Source:** Park and Sandhu (2002)

In 2004, a very similar model has been proposed by Zhang et al. in Zhang et al. (2004). In this model (Figure 2.10). They position the *usage decisions* at the center of the model with the objective to make it a more intuitive representation of the model UCON_{ABC}.

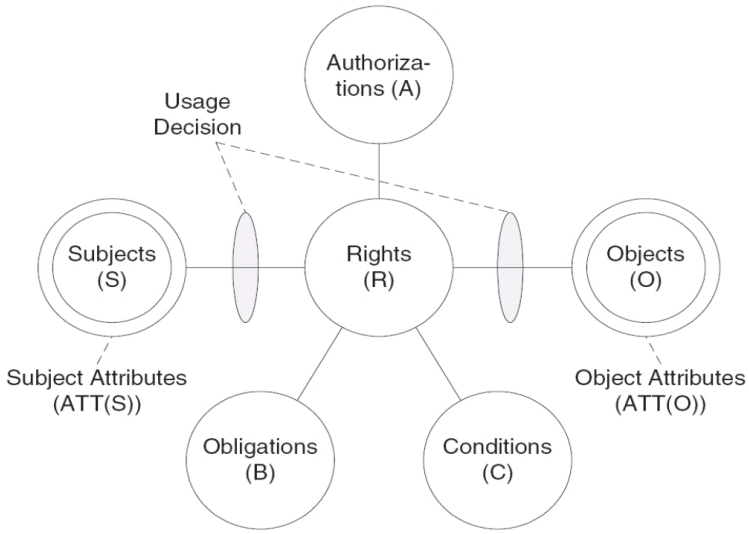


Figure 2.9: Usage Control model, **Source:** Sandhu and Park (2003)

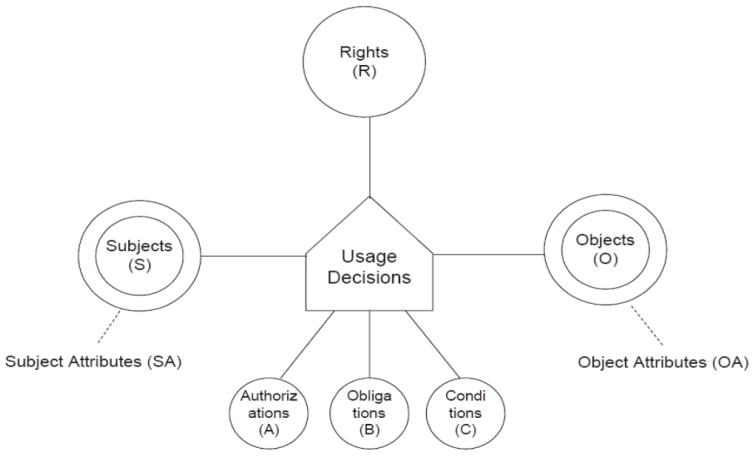


Figure 2.10: UCON alternative view, **Source:** Zhang et al. (2004)

The UCON model is composed of eight main elements: Subjects, Subjects attributes, Rights, Objects, Objects attributes, Authorisation, Obligations and Conditions. All these concepts exist at the application layer and represent concepts from the business layer.

- *The subject* has a unique identity or not. If he/she has one, an *accounting* related to its interventions may exist. Otherwise, the anonymity may be accepted and some attributes such as *prepaid credits* may be enough to provide rights. Three different subjects are defined: (i) the consumer subject (CS) which is for instance the subject that watches a DVD, (ii) the provider subject (PS) which is for instance the subject owner of the copyright over the content or the subject that provides the content, and (iii) the identified subject (IS) which is the subject identified by the object (which includes private information over the IS), e.g., the patient concerned by a medical record file.

- *The subject's attribute* provides complementary and mutable information related to a subject it is linked to. These attributes are e.g., a prepaid credit, a group name, a role, or a memberships.

- *The object* is the entity on which the subjects have rights. An object may be original or derived. Derived, generally, means a copy of an original to create a new object which includes at least a part of the original.

- *The object's attribute.* Objects also have attributes which means properties usable for access decision making. E.g., security labels and object's classes. Object's attributes may contribute to provide rights related to the use of an object, such as: value, permissions based on role or the amount of credits needed to access the object (e.g., 10 euros are necessary to access a DVD). A subject's attribute and an object's attribute also permits to include information such as the Access control list.

- *The rights* are privileges needed by the subject to access an object. Such as for the subject, rights are subdivided in three categories: customer right (CR), provider right (PR) and identified right (IR). In traditional access control models (such as DAC, MAC, RBAC), the access is provided almost systematically based on a matrix (e.g., the ACL) and this matrix permits to define the access rights as soon as they are requested by the subject based on the group (or the role) he/she is included in. The UCON model goes a step further while remaining compatible with those traditional access control models. The *usage decision* is made based on the subject's attributes, on the object's attributes, the authorisations, the obligations, and the conditions as well. The last three elements are the **A**uthorisations, the o**B**ligations and the **C**onditions. These elements have led to the naming $\text{UCON}_{\mathbf{ABC}}$.

- *The authorisation* is a functional attribute which must be evaluated for a *usage decision* and that returns if the subject is authorised to perform a rights request on an object, or not. authorisation evaluates a subject's attributes, an object's attributes, and the requested rights regarding a set of authorisation rules for the *usage decision.* These authorisations may be (Figure 2.11, extracted from Sandhu and Park (2003)) pre–authorisation (preA) if they are performed before the utilisation of the requested rights or ongoing–authorisation (onA) if they are performed during the utilisation of the requested rights. Certain authorisations may request subject's attribute or object's attributes updating. This update may be before, during or after the usage.

- *The obligation* is a functional attribute which verifies if the subject satisfies certain conditions before or during the usage. The obligation, such as the authorisation, may be

pre–obligation (preB) or ongoing obligation (onB). A preB is for instance the duty of a subject to fulfil some personal information before being granted access to a subject. The obligation may or may not use subject's attributes or object's attributes, not for decision making, but to select which obligation to apply. Additionally, it could be necessary for these obligations to update these attributes.

- *The condition* is a decisional factor based on the environment and oriented on the system. As the obligation, subject's attributes and object's attributes may be used to select which conditions are being used. Nevertheless, contrary to the authorisation and the obligation's variables, the condition's variable does not evolve because it is not under the control of the subject. Equally, the evolution of the condition may not update subject's attributes or object's attributes. The conditions are e.g., the local time, the temperature, or the location.

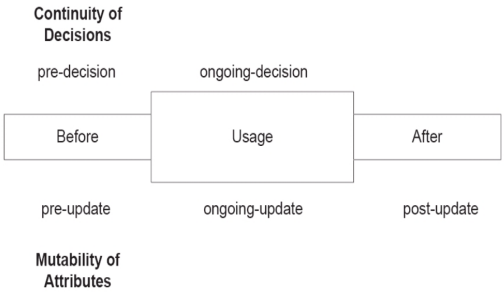


Figure 2.11: Continuity and mutability properties of UCON, **Source:** Sandhu and Park (2003)

The notion of context has also been addressed by Covington et al. (2001) who proposes a solution to integrate the contextual notion at the role (according to the RBAC model) and which defines a *Environment Role*.

2.3 Roles and rights engineering methods for business/IT alignment

Two types of approaches coexist regarding the roles and rights engineering methods: the *top–down* and the *bottom–up* (Kern et al. (2002)). This state of the art related to the roles and rights engineering methods have been restrained to the analysis of *top–down* solutions which exploit the concepts existing at the application or business layers of the organisation. This means that the other solutions such as those based on roles mining have not been considered. These solutions aim, according to Vaidya et al. (2007), at *utilising the existing permission assignments to formulate roles. Starting from the existing permissions before RBAC is implemented, the bottom–up approach aggregates these into roles.* This restriction is mainly justified by the fact that the *top–down* approaches traditionally do not recognise the existing permissions although the *bottom–up* does not consider business concepts from the organisation, which is in opposition to our research objective regarding the enhancement of the business/IT alignment.

To realise this state of the art related to the t*op–down* methods for roles and rights, we have firstly analysed the existing states of the art (Epstein and Sandhu (2001), Crook et al. (2003) and Crook et al. (2005), Fuchs et al. (2011)) and we have systematically reviewed the methods presented in the same sources of information as for the state of the art of the access control models. In this section, we present the methods which we have estimated more suitable to motivate the objectives of our research.

The next subsections introduce these frameworks and their particularities and characteristics as well.

2.3.1 Role/Permission Assignment Model

R/PAM (Role/Permission Assignment Model) is a model proposed by Epstein which permits to demonstrate that it is possible to decompose roles into permissions or to aggregate permissions into a role (Epstein (2002)). To perform this, Epstein introduces three new layers between the Role and the Permission to complete the Permission–Role assignment from RBAC. Therefore, the role is decomposed into jobs, the jobs are decomposed in workpatterns, and the workpatterns are decomposed in atomic tasks which are associated to permissions. Epstein cites in his work three approaches against which he compares his method, the Role–finding approach, the Napoleon approach and the Access control of the healthcare information system.

- Roeckle et al. (2000) proposes the Role–finding approach, a method based on a process–oriented approach to define roles. In this method, the authors use a metamodel built upon three layers: processes, roles and access rights. The process layer includes the concepts of job function, job position, organisational unit, but also the information system and security system which supports and protects it. The concepts from the role layer are the role, the sub–roles, and the bundles of rights. The concepts from the access rights layer are the groups and subgroups. Roeckle et al. (2000) do not define more profoundly these concepts. The metamodel aims at representing the relations which connect the concepts with each others, with the layers and between the layers. A procedural model is defined in parallel to the metamodel. The latter expresses the steps for instantiating the concepts of the process layer.

- The Napoleon model is issued from the work of Thomsen et al. (1999). The objective of this model is to aggregate permissions of roles by the intermediary of policies that are divided in three groups which allow their engineering. These groups are (1) the local policies where users are assigned to role, (2) the application policies where application encapsulate application specifications and (3) the semantic policies where the upper application policies are combined, together with the constraints and other information, into a semantic layer which has for advantage to permit users not from the IT administration (and thus more from the business layer) to define policies.

- Chandramouli describes in 1999 a five steps methodology to define an access control service for an information system in the field of health. Thereafter, he defines a framework named Dynamic Authorisation Framework for Multiple Authorisation Types (DAFMAT) which is composed of an hybrid access control model and of a logic–driven authorisation engine (Chandramouli (2001)). The hybrid access control domain is composed of RBAC and DTE (Domain Type Enforcement). DTE defines a security context in a domain. The

hybrid access control model contains: authorisation entities (user, role, subject, domain and object–type), relations amongst authorisation entities (mapping between source and target authorisation entities) and constraints governing the relations (constraints over the mapping).

As the R/PAM method aims at extending RBAC regarding the assignment of permissions to roles, the concepts introduced by the method (jobs, workpatterns and tasks) are, such as RBAC, at the application layer, although they realise concepts from the business layer. This statement is corroborated by the analysis of the case study related to the duties of a university's *office administration* presented in Epstein (2002).

2.3.2 Analytical Role Modelling Framework

Crook et al. proposed in 2003 in Crook et al. (2003) and in Crook et al. (2005) a framework to model roles following the RBAC model together with the definition of the links with the organisational structure. His work is based on the Mintzberg (1992) contribution which aims at associating the role with the organisational context regarding two dimensions: lines of authority and division of work. Thereby, Crook et al. consider that a role represents a position in an organisation, that the role has different levels of seniority and finally, that this role is operated in a organisational domain (the context in which the role is played). Roles are classified according to the three following categories that are integrated in the framework: *Roles based on seniority*, *Roles based on function* and *Roles based on market*.

ARMF is composed of two levels: a *meta–level* that includes role, asset category, context type, operation and access policy (in plain line on Figure 2.12). The definition of role is refined from the RBAC definition with the notion that roles are *a way of defining positions in organisations, bundling responsibilities, or perhaps representing a qualification*. The asset category allows categorising the information to determine the access policy which applies. The context is not formally defined but it is required by some policies in order for them to be resolved. The operation is an operation that may be performed on an asset. These conceptual components allow defining access policies considered through the relationship between the roles sets, the sets of operations and the asset category.

The second level is the *Instance level* in dash lines and it defines the users, the context instance, the assets instance, the role instance and the operation request. The user is *an instance of an assigned role*. The operation request aims at modelling the request made by a user wishing to execute an operation. The context instance and the asset instance are respectively instance of the context and the asset.

From the ARMF principle that to access an asset, a user must at least be assigned to one of the three roles: functional, seniority and contextual, an access control policy can be defined from the model of Figure 2.12. The advantages of this modelling approach are that it enables the user with different roles or responsibilities to access the same functions (not represented on the metamodel). Additionally, ARMF permits to define contextual roles by defining the existing links between the assets and contexts, and between the contexts and the roles. This is illustrated in Figure 2.13: on the left, when a role is a contextual role, there exists an association between this role and the context concept on hand and between the context and the asset category on the other hand. On the right, the picture from left is instantiated regarding the access to the

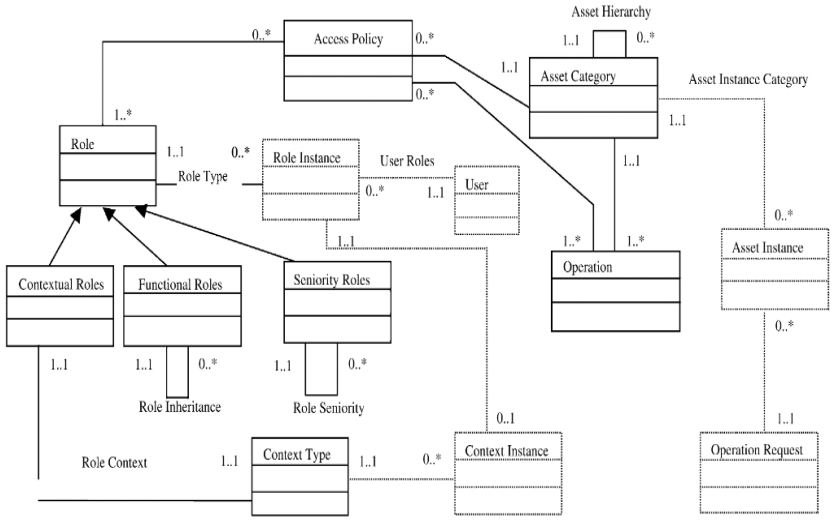


Figure 2.12: Key components of ARMF framework, **Source:** Crook et al. (2005)

nurse record in the context of a ward or not. The formalism of the graphic notation used to represent the policy scenarii diagram of Figure 2.13 has been developed by the author.

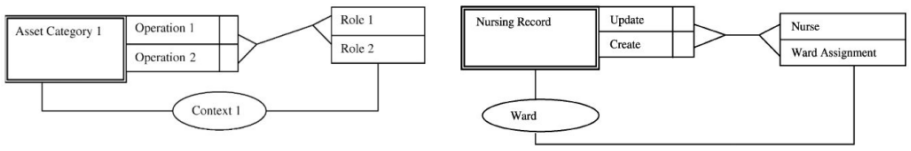


Figure 2.13: Composition of a policy diagram, **Source:** Crook et al. (2005)

In 2005, Crook et al. (2005) exploits i* to model access policies that take into account the organisational context. The i* framework permits to model the relations between actors using the Strategic Rationale (SR) model proposed by Yu and Liu (2000) (Figure 2.14).

Crook et al. (2005) make the link between the RBAC model and the SR model from i* framework to derive the roles from the actors and the permissions from the tasks. An actor from i* may be an agent, a role or a position. An agent is a physical entity such as a human (e.g., a *family doctor* in Figure 2.14) . A role is defined as an abstract actor that may be chosen by an agent, and a position is a set of roles that can be assigned to an agent.

2.3.3 Uses cases

Fernandez and Hawkins (1997) explain that one method for determining functional requirements is the definition of Uses cases. The users of the system are interviewed to express how they interact with the system. Fernandez et al. propose a method to determine the needs for a role

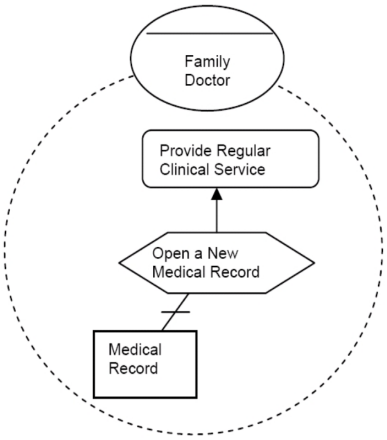


Figure 2.14: Strategic Rational diagram, **Source:** Yu and Liu (2000)

considering the use cases and the sequences of the use cases. Afterwards, a security administrator defines the authorisation rules based on all the use cases of the system. The use cases are described using: a title, the actors (who may be the roles, the users or the other systems), the pre–conditions, the descriptions, the exceptions, and the post–conditions.

The access rights to specific objects are then derived from the use cases. The authorisation rules may take the form (S, O, T, P) with S: the subject, O: the object, T: the type of permitted access, and P being an optional constraint. The permitted access and the subject represent an access right and an actor which exist at the business layer (the latter is defined by Fernandez et al. by a *materials employee*). The object is a concept which exists at the application layer.

The use cases only permit to define functional specifications. Fernandez et al., therefore, propose an extension of the use cases to tackle non–functional requirements (e.g., security). Therefore, they use stereotypes, or meta–classifications of UML elements. Stereotypes of the system architecture requirements are for example: load, fault tolerance or security.

At the application layer, the necessary permitted access T is calculated from the use cases by considering the methods to be invoked by the actor. As a consequence, if in a scenario diagram, an actor interacts with objects through methods, then the use cases provide a set of formal authorisation rights (R) that are expressed, according to Fernandez et al. by to the triplet: R (A, M, O) with A: the Actor, O: the Object and M: the Method. According to the authors, tools may be used like Rational Rose or Paradigm +, to represent use cases and to that extent, to generate the required permissions rules.

2.3.4 Scenario–driven role engineering

Neumann and Strembeck (2002) propose a role engineering method based on the scenarii model. A scenarii is a subset of a task (a business activity such as processing an order) and this task is a subset of a *work profile* (the set of tasks that an employee can perform). Additionally, the scenarii can be considered as a set of steps on which particular access operations are associated.

By consequence, a subject which operates a scenarii must possess all the permissions necessary to realise each steps of this scenarii.

Neumann and Strembeck define a scenarii–driven role engineering process in seven major activities, Figure 2.15:

1. *Identify and model usage scenarii.* This activity aims to identify and model, using scenarii, the sensitive system usage.

2. *Derive permissions from scenarii.* This step aims to identify and store the access operations necessary to execute the sequence of each the scenarii.

3. *Identify constraints.* The constraints which need to be enforced on permissions are identified, e.g., separation of duties, cardinalities or time–dependencies.

4. *Refine scenarii model.* This step aims at reviewing the set of scenarii to identify the similarities and for generalisation if possible.

5. *Define tasks and work profiles.* Tasks are defined based on composing scenarii and according to the constraints. The scenarii may compose different tasks and the latter are used to elaborate the work profiles.

6. *Derive preliminary role–hierarchy.* A preliminary role–hierarchy is created using the role profiles and the permissions.

7. *Define the RBAC Model.* Concretely the RBAC model is elaborated using the preliminary role–hierarchy, the permissions, and the constraints.

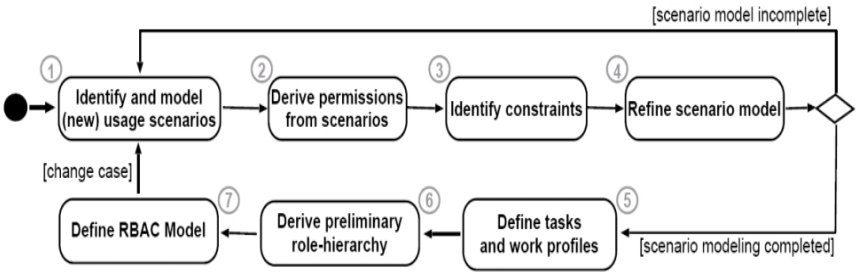


Figure 2.15: Scenario–driven role engineering process, **Source:** Neumann and Strembeck (2002)

A set of documents is issued from these activities: scenarii model, permissions catalogue, constraints catalogue, tasks definitions, work profile and the RBAC model. The RBAC model is the final result of the role engineering process and includes all the roles of the system arranged according to one or more hierarchy(ies). The Figure 2.16 shows the interrelation between the model and the document extracted along the scenarii–driven role engineering process. This picture highlights the concepts of scenarii, permission, constraint, task and work profile exist at the business layer although the RBAC model is at the application layer.

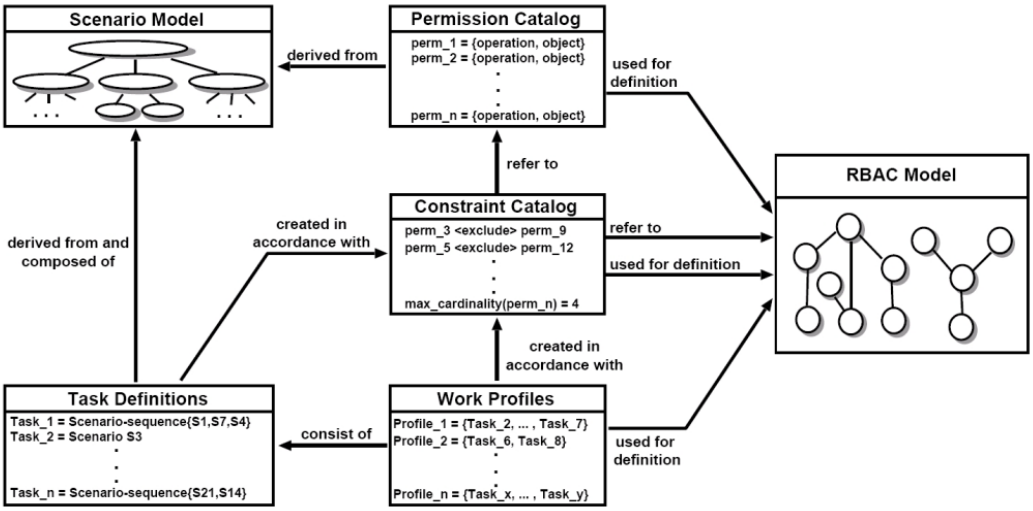


Figure 2.16: Interrelation between the model and the documents, **Source:** Neumann and Strembeck (2002)

2.4 Conclusions

In this chapter, we have firstly reviewed the state of the art (summarised in Table 2.1) related to the access control models (**Item 4** of Figure 1.4). The analysis of this state of the art shows that the layer the most concerned by the access control models is the application layer.

The first developed models (MAC, DAC) generally consider the artefacts of subject which gains or not access rights over the concept of object. All the concepts from these models exist at the application layer. The models reviewed afterwards (RBAC, ABAC, TBAC, TRBAC, and TMAC) associate the concept of subject through one, or with a set of, concept(s) that reflect the specificity of the models, e.g., the concept of role from the RBAC model which links the concept of user with the concept of permission or the concept of team from TMAC. These models tend to rely on the business layer to define their concepts. E.g., the concept of role which is defined based on *the authority and the responsibility conferred on the user*. However, these concepts remain exploited at the application layer, and focuses especially on the assignment of permissions related to actions on data objects or files. TRBAC goes one step further and considers that the permission is associated to the subject based on the tasks that he has to perform. However, this task is associated to a role which does not correspond to the business role but rather to a role which gathers users with an identical need of set of permissions. OrBAC tends to clarify, enhance and basically formalise the association between the concrete and the abstract levels, as explained in Cuppens and Miège (2003). To that end, OrBAC considers one concept of organisation to elaborate concrete access rights policies.

Afterwards, we have reviewed the state of the art in the field of the business/IT alignment methods related to the rights and roles engineering methods (**Item 3**). This state of the art highlights methods which also tend to engineer the access rights with the application layer and considering specific access control models. These methods most often exploit the functional re-

quirements analysis (e.g., ARMF, the scenarii–driven role engineering method) and are mostly dedicated to the definition or the instantiation of the RBAC model (e.g., the role–finding approach, the Napoleon approach, ARMF or the scenarii–driven role engineering method).

Model	***Concepts* – Layer of existence**
MAC	*Subject, object, action* – All concepts exist at the application layer.
DAC	*Subject, object, action* – All concepts exist at the application layer.
RBAC	*User, role, object, action, permission, role hierarchy, SOD, session* – User is defined as a human and role is defined based on the job functions and responsibilities. Nevertheless, all concepts of the model are exploited at the application layer.
ABAC	*User, subject, permission, attribute* – All concepts exist at the application layer except the user, at the business layer, which is realised by the subject.
TBAC	*User, task, permission* – User and task are defined based on the business layer but are exploited at the application layer to calculate permissions at that same layer.
TRBAC (Task–Role)	*User, role, task, permission* – User, role and task are defined based on the business layer but are exploited at the application layer to calculate permissions at that same layer.
TMAC	*Team, user, object, permission* – Team and user represent elements from the business layer but which exist at the application layer. Object and permission exist at the application layer.
TRBAC (Temporal)	*User, temporal information, role, object* – The user and the temporal information are directly derived from the business layer and are represented at the application layer. Object and permission exists at the business or at the application layer.
OrBAC	*Role, activity, context, view, subject, action, object, organisation* – The concepts from the concrete level correspond to concepts from the application layer and the concepts from the abstract level correspond to concepts from the business layer.
UCON	*Subject, subject attribute, right, object, object attribute, authorisation, obligation, condition* – All these concepts exist at the application layer and represent concepts from the business layer.
Method	***Concepts* – Layer of existence**
R/PAM	*Idem RBAC + job, workpattern, task* – The concepts exploited represent concepts from the business layer which are handled at the application layer.
ARMF	*Roles, asset category, context type, operation* – The concepts exploited represent concepts from the business layer which are handled at the application layer.
Use cases	*Subject, object, type of permitted access, constraint* – All concepts exist at the application layer
scenarii–driven	*Idem RBAC + scenarii, permission, constraint, task* – The concepts which define the scenarii and the permissions are at the business layer. The concepts of RBAC are at the application layer

Table 2.1: State of the art summary

In summary, throughout this state of the art, we notice that the objective of the models and methods is mainly to support the IT managers in defining, deploying and managing the access rights on information system. Thereby, we have observed the five following points:

- the concepts from the business layer are used sporadically in early access control models such as MAC and DAC,
- RBAC has brought a significant enhancement in the alignment of the IT with the business by considering the concept of role,
- the models that have succeeded RBAC are getting closer to the business layer (Task–Role and Organisation Based Access Control) and, thereby, tend to deeply consider the employees' obligations and responsibilities regarding the tasks that they are assigned to,
- the rights engineering methods follow this trend and tend to equally consider the concepts from the business layer (such as ARMF, R/PAM). Yet, none of them has addressed this issue through the notion of employee's responsibility.
- up to date, although the new arising governance needs, no model allow a full alignment between the business layer, where rights should be engineered, and the application layer, where the latter should be assigned to users.

Publication related to this chapter:

- C. Feltus, Preliminary Literature Review of Policy Engineering Methods – Toward Responsibility Concept, in *Proceedings of the International Conference on Information and Communication Technologies: from Theory to Applications (ICTTA)*, Damascus, Syria, 2008. IEEE.

Chapter 3

Needs of governance and fundamentals of responsibility

3.1 Introduction

In the previous chapter, we have observed a trend of the access control models and rights engineering methods towards a wider exploitation of the business concepts and of responsibility for the management of access rights.

In parallel to this evolution of the access rights management, as explained in Section 1.2, many governance standards and norms have recently appeared to improve the corporate governance, the IT governance and the business/IT alignment (also named strategic alignment). In this chapter, we deepen the link between the new governance needs and the observed trend for the management of the access rights. Therefore, in Section 3.2 we give an insight on the governance and business/IT alignment by depicting a set of main representative definitions coming from the academic and professional worlds. Then, in Section 3.3, we analyse governance frameworks to explore how the business concepts and the responsibilities are considered as new needs to be focused on for the access rights management. To that end, the documents analysed are not limited to the field of IT (**Item 1** of Figure 1.4) but also include documents from a more wide range of organisational layers and domains like finance and accounting (**Item 2**). The output of this analysis constitutes an unrefined picture including significant zones of meaningful concepts to be considered by the access rights management, and motivated by the governances frameworks.

In Section 3.4, the existing access control models and rights engineering methods, reviewed in Chapter 2, are analysed in order to figure out how they, to date, integrate these zones of concepts.

Finally, based on this analysis, we acknowledge the importance of having the responsibilities suitably defined and assigned to employees. To consider this responsibility in a business/IT alignment approach, a primary step consists in discovering the fundamentals of responsibility to better apprehend it. This review is achieved in Section 3.5.

3.2 Governance insight

This section aims at providing an insight of the corporate governance, the IT governance and the business/IT alignment. This is achieved by reviewing the main definitions of the concepts and by understanding how they are articulate with each other.

3.2.1 Corporate governance

This section aims at introducing the concept of governance. Therefore, we have selected three definitions provided by the international organisations OECD (2004), IFC (2002) and The World Bank (1991):

- OECD Corporate Governance (OECD (2004)) defines the corporate governance as: *a set of relationships between a company's management, its board, its shareholders and other stakeholders. Corporate governance also provides the structure through which the objectives of the company are set, and the means of attaining these objectives and monitoring performances are determined. Good corporate governance should provide proper incentives for the board and management to pursue objectives that are in the interest of the company and its shareholders and should facilitate effective monitoring.*

- The International Finance Corporation (IFC (2002)) provides the following definition: *Corporate governance refers to the structures and processes for the direction and control of companies. Corporate governance concerns the relationships among the management, the Board of Directors, the controlling shareholders and other stakeholders. Good corporate governance contributes to sustainable economic development by enhancing the performance of companies and increasing their access to outside capital.*

- The World Bank (1991) defines it by: *the manner in which power is exercised in the management of a country's economic and social resources for development.*

The definition of the OECD is completed by The Open Group (2006) with the idea that the corporate governance focuses on the *rights, roles, and equitable treatment* of shareholders as well as on the transparency and the responsibilities of the board which are to scrutinise and accompany the corporate strategy on the one hand, and to define and control the performance of managers on the other hand. According to this, the corporate governance also contributes to define the accountabilities of the board, towards the shareholders, regarding the company.

Based on the above definitions, we acknowledge that the corporate governance is a very generic term that aims at supporting the definition, at directing and at monitoring the achievement of the high level objectives of the company. To that end, the corporate governance aims at sustaining the relation between the management, the board of directors, and the shareholders. It also expresses the decision making policies related to corporate issues with the aim to ensure the adequacy of the resources usage according to the strategic objectives of the organisation.

Nowadays, the development of the information systems has created situations where it is no longer conceivable to consider dealing with corporate governance without exploiting information technology solutions. As a result, the connections between the corporate and the IT governance have been strengthened and are actually perceived as indissociable. Indeed, as explained in the next section, the corporate governance defines and monitors the performance of the IT

governance, and the latter provides new corporate opportunities that influence the corporate strategy.

3.2.2 IT governance

IT governance is a subset of corporate governance (Parkes (2004)). IT governance is defined in international standards such as ISO38500 (2008), in academic literature such as the MIT report (MIT (2002)) or in professional frameworks like IT Governance Institute (2003):

- From the ISO/IEC (ISO38500 (2008)): *Corporate Governance of ICT is the system by which the current and future use of ICT is directed and controlled. It involves evaluating and directing the plans for the use of ICT to support the organisation and monitoring use to achieve plans. It includes the strategy and policies for using ICT within an organisation.*

- From the ITGI (IT Governance Institute (2003)): *IT Governance is the responsibility of the board of directors and executive management. It is an integral part of enterprise governance and consists of the leadership and organisational structures and processes that ensure that the organisation's IT sustains and extends the organisation's strategies and objectives.*

- MIT Sloan Centre for Information Systems Research (MIT (2002)) proposes: *IT Governance is specifying the decision rights and accountability framework to encourage desirable behaviour in the use of IT.*

- The survey of the literature by academics from the University of Tasmania results in the following definition (Webb et al. (2006)): *IT Governance is the strategic alignment of IT with the business such that maximum business value is achieved through the development and maintenance of effective IT control and accountability, performance management and risk management.*

Amongst the IT governance definitions from the academic research, two of them hold our attention. The first one is from Weill and Ross (2004) saying that *IT governance is about specifying the decision rights and accountability framework to encourage desirable behaviour in the use of IT*. The second one is from Broadbent (2002) that focuses on the usage, by the authority in an organisation, of the IT to reach the goals of this organisation.

Based on the above definitions, we observe that the objectives of the IT governance are equivalent, at the IT level, to the objectives of the corporate governance. Hence, we note that IT governance aims at defining, directing and monitoring the IT strategy, that it is the responsibility of the board of directors, and that it specifies the decision rights and accountability framework.

For the IT Governance Institute (2007), the IT governance, as explained in Figure 3.1, is composed of five dimensions: *Value Delivery* (IT supports the business and allows maximising the benefits), *Risk Management* (risks are managed properly), *Resource Management* (IT resources are used reasonably), *Performance Measurement* (objective achievements are follow up), and *Strategic Alignment* (IT is aligned with the business). Hence, the strategic alignment is considered, by COBIT 4.1, as a constituting element of the IT governance. This link between the IT governance and the strategic alignment is reviewed in the next section.

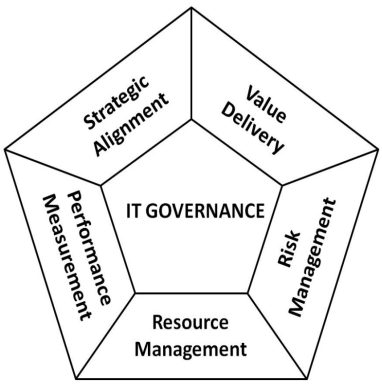


Figure 3.1: COBIT IT Governance focus areas, **Source:** IT Governance Institute (2007)

3.2.3 Business/IT alignment

Through academic research, the business/IT alignment has been defined as *the degree to which the missions, objectives, and plans containing in the business strategy are shared and supported by the IT strategy* (Reich and Benbasat (1996)). The authors consider the alignment at the strategic layer but did not consider the application layer. Henderson and Venkatraman (1993) have addressed this alignment among the four following components: business strategy, IT strategy, business infrastructure and IT infrastructure. They consider four types of alignments: strategy execution, technology potential, competitive potential and service level (Figure 1.2). Luftman (1996) defines the business IT/alignment by *the extent to which the IS strategy supports, and is supported by, the business strategy.* Additional definitions are proposed in the state of the art from Chan and Reich (2007). A holistic view of the researches in this field allows arguing that the main motivation for aligning the business with the IT is the enhancement of the performance of the company. More recently, Beimborn et al. (2009) extended this view and proposed a theoretical model that links the alignment with the governance mechanisms and with the business process performances and Maes et al. (2000) proposes a unified framework for the business/IT alignment and analyse its integration with architecture framework.

Business/IT alignment is sometimes perceived as a part of the IT governance, and other times, it is perceived as a domain to be handled disjointedly. In COBIT 4.1 and in Webb et al. (2006), amongst others, the strategic alignment is a composing element of the governance. The strategic alignment is defined for COBIT (Section 3.3.1.1) by *ensuring the linkage of business and IT plans; defining, maintaining and validating the IT value proposition; and aligning IT operations with enterprise operations* and is an *IT Governance Focus Areas* (Figure 3.1). The Open Group (2006) equally argues that the *IT governance provides the framework and structure that links IT resources and information to enterprise goals and strategies [...] to ensure that the enterprise's IT assets support its business objectives.* In the ISO38500 (2008) standard, as will be described in Section 3.3.1.3, the relation between both is made clearer: *corporate governance of IT assists directors to ensure that IT use contributes positively to the performance of the organisation, through alignment of IT with business needs.* In this standard, business/IT alignment is considered as a subset of the corporate governance of IT.

A few other perspectives exists regarding the association between the IT governance and the

business/IT alignment. For Haes and Grembergen (2008), the goal of IT governance is achieving a better alignment between the business and IT. They define, in De Haes and Van Grembergen (2004), the IT governance as *the organisational capacity exercised by the Board, executive management and IT management to control the formulation and implementation of IT strategy and in this way ensure the fusion of business and IT*. Another perspective is proposed by Ridley et al. (2004) who explain that, for an efficient governance of IT, it is fundamental to have business/IT alignment.

In the thesis, we retain this statement that business/IT alignment is part of the IT governance. Moreover, we consider in this research the alignment of the business concepts from the business layers of the organisation (**Item 6** of Figure 1.4) down to application concepts from the application layer (**Item 5**), and more specifically down to the access rights policy (**Item 7**). This alignment approach corresponds to the *strategy execution perspective* from Henderson and Venkatraman (1993).

3.3 Governance frameworks

The goal of this section is to explore how the business concepts and the responsibility are considered, by the governance frameworks, as new needs to be taken up by the access rights management.

Therefore, we have selected five frameworks that are assumed to be the most representative according to the Interim Report from the ISO Study Group on ICT Governance (Report (2007)). Among others, this Interim Report aims at identifying the ICT governance's needs to be addressed in the standard ISO38500 (2008). Therefore, an analysis of the existing frameworks was achieved through a review of the governance activities in the countries involved in this report, including: USA, Japan, Australia and New–Zealand, South Africa, Russia, Singapore, United–Kingdom, Belgium and Luxembourg. During this review, most of the countries' representatives stressed the importance of the following frameworks: COBIT, ISO/IEC 27000, SOX and Basel II to address the governance in their respective countries. COBIT and ISO/IEC 27000 are dedicated to the IT governance although SOX and Basel II are dedicated to the corporate governance. Others frameworks were also cited but have not been included in the review because they are less frequently mentioned or because mostly they concern the technical layer. This is the case, among others, of ISO9000 (2000), ITIL (2001), or ISO15504 (2004). The relevance of the selected frameworks has, furthermore, been corroborated at the scientific level by Spremić (2011) and Spremić and Spremić (2011).

The set of frameworks and norms analysed in the sequel of this section include, as a consequence, on one hand COBIT, ISO/IEC 38500 and ISO/IEC 27000. The ISO/IEC 38500 standard has been added in this list since it aims to be a generic standard that provides high level principles for the governance. These three frameworks introduce the concrete needs of governance for the access rights that motivate our research. On the other hand, we have analysed Basel II and SOX to highlight how the high level and concrete needs are considered in standards and norms which address the governance of the financial and accounting sector.

In the next sections, these frameworks are analysed to understand their contributions for the alignment of the business layer down to the application layer. The needs that we target

contribute to improving the assignment of access rights required by an employee to achieve a task. This assignment is a process that is defined through inputs (e.g., business information), outputs (e.g., access rights rules), activities (e.g., define the task and the rights necessary, assign the rights) and actors (e.g., the employee that achieves a task, the access rights manager, the business process owner) (Lindsay et al. (2003)).

3.3.1 IT governance framework

The three IT governance frameworks analysed are Cobit, ISO/IEC 38500 and ISO/IEC 27000.

3.3.1.1 COBIT

The scope of COBIT is the Governance of IT. This framework (IT Governance Institute (2007)) enables the development of clear policies and good practice for IT control throughout enterprises. COBIT Executive Overview describes COBIT as follows: *COBIT is a framework and supporting tool set that allows managers to bridge the gap with respect to control needs, technical issues and business risks, and communicate this level of control to employees. COBIT enables the development of clear policies and good practices for IT control throughout enterprises. COBIT is continuously kept up to date and harmonised with other standards and guidance. Hence, COBIT has become the integrator for IT good practices and the umbrella framework for IT governance that helps in understanding and managing the risks and benefits associated with IT. The process structure of COBIT and its high–level, business–oriented approach provides an end–to–end view of IT and the decisions to be made about IT.*

COBIT processes are linked to the strategical, the management and the operational aspect of the information security governance.

COBIT 5 (IT Governance Institute (2012)) was released in June 2012 and is aligned with frameworks and standards such as Information Technology Infrastructure Library (ITIL (2001)), International Organization for Standardisation (ISO38500 (2008) and ISO27000 (2012)), and The Open Group architecture framework (The Open Group (2006)).

3.3.1.2 Concrete COBIT governance needs related to the access rights management

Access Control is defined by COBIT 4.1 as *the process that limits and controls access to the resources of a computer system; a logical or physical control designed to protect against unauthorised entry or use.* The management of access rights is integrated in the Control Objectives DS5.3 Identity Management and DS5.4 User Account Management from the control DS5 Deliver and Support, Ensure Systems Security.

In both of these controls, COBIT requests:(i) the users to be uniquely identified, (ii) the users' access rights to the systems and to data to be in line with defined and documented business needs and that job requirements are attached to users' identities, (iii) the users' access rights to be requested by user management, approved by system owners and implemented by the security officers, (iv) the user identities and access rights to be maintained in a central repository, (v) the rights and obligations related to access the enterprise systems and information to

be contractually arranged for all types of users, (vi) these procedures to apply to all users, including the administrators (privileged users) and the internal and external users, for normal and emergency cases and(vii) to detect and resolve unauthorised accesses. Additionally, the PO2.3 control provides last requirements to know: (viii) That the data must be classified for access control.

As already noted in Section 3.2.1, COBIT considers that having the responsibilities appropriately defined for each employee is a high level need of governance. A deeper analysis highlights that COBIT also addresses the responsibilities of all the business roles played by the employees involved in the IT governance actions and is formalised through a RACI chart matrix attached to all 34 COBIT processes. The RACI acronym stands for Responsible, Accountable, Consulted and Informed. These responsibilities are assigned, through the RACI matrix, to predefined business roles, themselves later played by employees. We have analysed COBIT and propose a summary in the form of a metamodel (Figure 3.2). The four responsibilities of the matrix are presented in detail in the following since they are going to be significant for the sequel of our research. To figure out the meaning of each responsibility, the latter are illustrated based on examples from the healthcare domain.

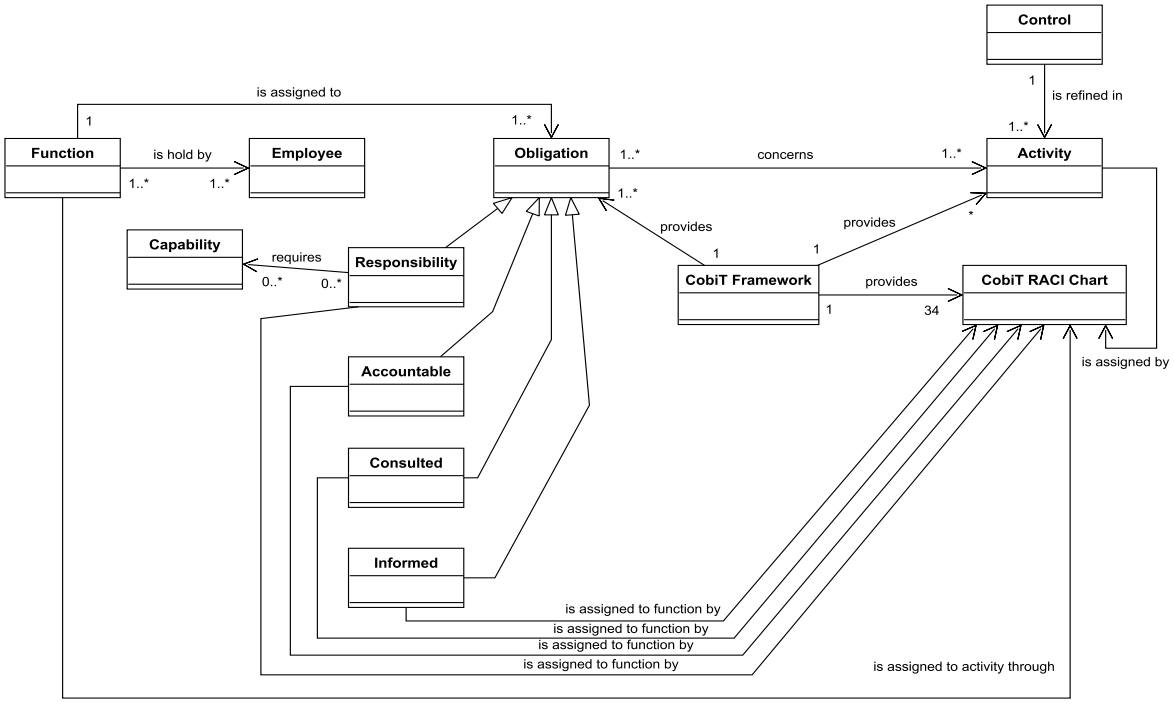


Figure 3.2: COBIT responsibility UML diagram

1. **Responsible:** The obligations of the role Responsible are to correctly execute processes or activities. The responsible often reports about the performance of the activity achievement to the employee who is accountable of deciding for that activity.

   *An example of **responsible** from the healthcare domain is the obligation of a doctor to perform the treatment of a patient, or is it the obligation of a secretary to prepare the*

order entry of a patient in the appropriate medical speciality and the justification of its realisation or the reason of a delay, or why some information is missing, and so forth.

2. **Accountable:** The obligation of the role Accountable is to own the responsibility over the quality and the end result of the process. This role directs and makes authorisations concerning a process. The accountable has obligations to report the achievement of the activity to the board of directors or to a governmental authority.

   *An example of **accountable** in the healthcare domain is the obligation for a doctor regarding the decision of the type of drug to provide to a patient and the justification of that drug towards that patient. It may also be, for instance, the decision not to give access to confidential data of one of its patient to a colleague.*

3. **Consult:** The obligation of the role that is consulted is to provide advice which allows the realisation of a business activity. The responsible has obligations but most of the time, his accountability is limited and the reporting is not mandatory.

   *An example of **consult** in the healthcare domain is when a young doctor asks for advice to a senior doctor. In that case, the senior doctor that accepts the role of advisor is obliged to give the correct information to the young doctor and to justify his advice towards an authority.*

4. **Informed** The role Informed receives information about process execution,the actions performed, the decisions made, and quality, and with whom there is one–way communication.

   *An example of **informed** in the healthcare domain is when a doctor in charge of a patient is informed about the care that has been provided to its patient, during the night, by the nurse on duty.*

COBIT also stresses the importance of the rights and capabilities. Rights appear in the sense of the access rights like in DS5.3 *Identity Management* or in the sense of rights and obligations linked to a contractual engagement as e.g., in AI5.4 *IT Resources Acquisition*. The capability is defined as: *Having the needed attributes to perform or accomplish [...]* and is related to a process or to an employee's responsibility. This requirement is at very high level and the required capabilities are not further identified in the framework. However, we interpret these capabilities as what is necessary for an employee to perform an action. These capabilities are illustrated, by COBIT, as follows:

1. ME1.5 Board and Executive Reporting: *Provide management reports for senior management's review of the organisation's progress toward identified goals [...].* In this first case the management report is one capability necessary for the senior management's review.

2. AI4.2 Knowledge Transfer to Business Management: *Transfer knowledge to business management which allows them to take ownership of the system and data and exercise responsibility for service delivery and quality, internal control, and application administration processes[...]* In this second case the knowledge is a capability necessary for the business management.

3.3.1.3 ISO/IEC 38500:2008

The international standard ISO/IEC 38500 (ISO38500 (2008)) is a high level framework that provides guidance on the role of governing bodies. It has been produced after a fast track of the Australian Standard AS8015 (2005). As consequence, ISO/IEC 38500 has been published after AS8015 and includes the same information, structured in the same way. The standard provides a set of six high level principles for the managers of the company to help them in evaluating, directing and monitoring the use of the information system of the company. The six principles are:

- Principle 1: Responsibility – employees understand and accept their responsibilities,
- Principle 2: Strategy – ICT plan fits and supports corporate plans,
- Principle 3: Acquisition – acquisitions are made for approved reasons,
- Principle 4: Performance – ICT supports the organisation and its evolutions,
- Principle 5: Conformance – ICT comply with external rules and internal policies,
- Principle 6: Human Behaviour – ICT meets the needs of the people.

ISO/IEC 38500 requests the directors to govern the information technology through three main activities: evaluate, direct and monitor (Figure 3.3, extracted from ISO38500 (2008)).

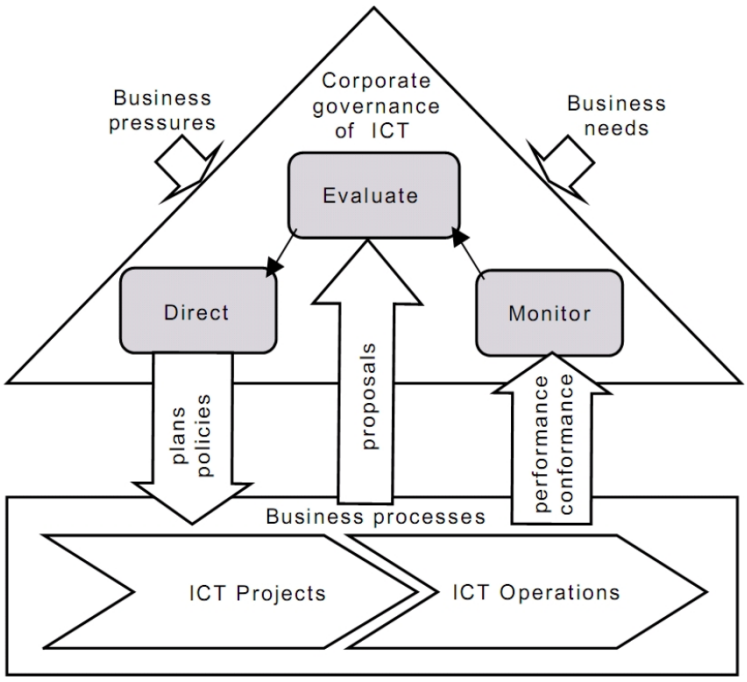


Figure 3.3: ISO/IEC 38500: Model for corporate governance of ICT, **Source:** ISO38500 (2008)

3.3.1.4 Concrete ISO/IEC 38500 governance needs related to the access rights

Because ISO/IEC 38500 is a very high level framework, it does not directly address the access rights management.

Notwithstanding, the framework acknowledges that the company include groups or communities of humans, each with their own needs, aspirations and behaviours. These needs are the accessibility and the ergonomics for the people using the IT. Additional needs are, for instance, the communication, the training, the reassurance, the working conditions and the development of skills.

Although the access rights are not the focus of the framework, it stresses the importance to consider the employee's responsibilities and accountabilities, especially the Directors personal accountability regarding a list of possible breaches of security standards, privacy legislations, social responsibility standards, environmental legislations and regulations, and so forth.

The standard also calls for the employees' assignment of *responsibilities to make decisions*. The employee assigned responsible must have the necessary competences and should, at the same time be held responsible for the business objectives and performances resulting to those decisions. Moreover, they must acknowledge and understand their responsibilities, and their performances should be monitored by the directors (first principle of the framework).

3.3.1.5 ISO/IEC 27000 family

The ISO/IEC 27000 serie (ISO27000 (2012)) constitutes an IT security management and governance framework. It aims at providing a set of best practices related to the information security management and the business/IT alignment since the processes aims at achieving business goals. The standard is operationalised by means of an overall information security management system (ISMS) that argues for continuous feedback and improvement steps. The serie covers IT or technical security issues such as, amongst others, the risk management, the implementation of security controls, the protection of the privacy and confidentiality. It is tailored to be applicable to organisations of all sizes. The standard gathers activities to be performed to manage the information security firstly and lists controls necessary to guarantee the corporate security objectives secondly. These controls are derived from ISO/IEC 27002 and are listed in the annex A of the framework.

Most of the management activities are enumerated in section 4 of the framework and are structured following the PDCA (Plan–Do–Check–Act) model. These activities concern the establishment, the implementation, the monitoring and review, and the maintenance and improvement of the ISMS. The assignment of responsibilities for the achievement of these activities is superficial and incomplete provided that only the organisation or the management is identified as being responsible.

The operational controls of ISO/IEC 27001 are retrieved from the control objectives listed in ISO/IEC 17799:2005 clauses 5 to 15 and correspond to implementation advices and guidances on best practices. These controls correspond directly to security practices necessary to be deployed to ensure the corporate information system security. Typical operational controls are access control (A.11) (Figure 3.5), information back–up (A.10.5.1), capacity management (A.10.3.1) or information access restriction (A.11.6.1).

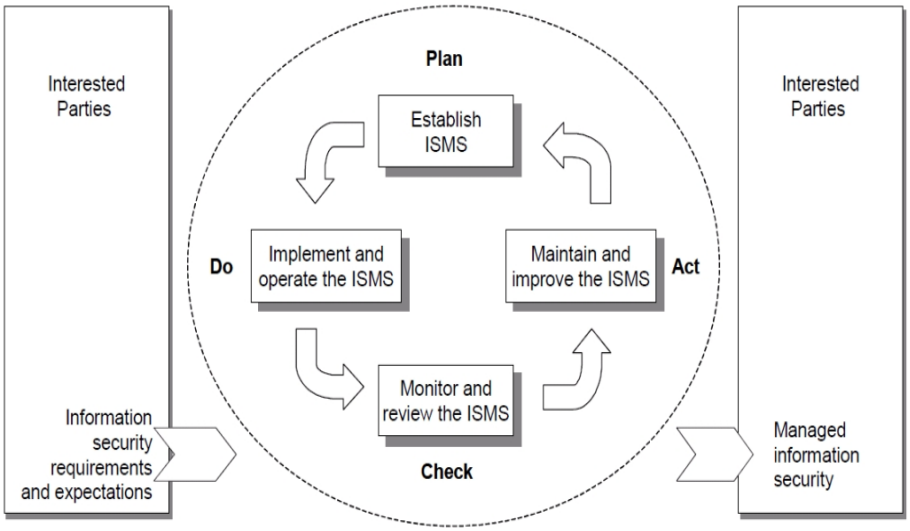


Figure 3.4: ISO/IEC 27001 PDCA model applied to the ISMS, **Source:** ISO27000 (2012)

A.11 Access control		
A.11.1 Business requirement for access control ***Objective: To control access to information***		
A.11.1.1	Access control policy	*Control: An access control policy shall be established, documented, and reviewed based on business and security requirements for access*
A.11.1 User access management ***Objective: To ensure authorized user access and to prevent unauthorized access to information systems***		
A.11.2.1	User registration	*Control: There shall be formal user registration and de-registration procedure in place for granting and revoking access to all information systems and services*
A.11.2.2	Privilege management	*Control: The allocation and use of privilège shall be restricted and controlled*
A.11.2.3	User password managment	*Control: The allocation of passwords shall be controlled through a formal management process*
A.11.2.4	Review of user access rights	*Control: Management shall review users' access rights at regular intervals using a formal process*

Figure 3.5: ISO/IEC 27001 Access Control, **Source:** ISO27000 (2012)

3.3.1.6 Concrete ISO/IEC 27000 governance needs related to the access rights

ISO/IEC 27000 defines the access control as *a means to ensure that access to information assets is authorised and restricted based on the business and security needs.* The access control may be *technical (logical), physical, administrative (managerial) or a combination of them.*

Additionally, the standard depicts the responsibilities of the managers. These responsibilities are mainly listed in sections 4.2 and 5.1. Both sections of the framework gather the activities to be done by the organisation at each of the four ISMS steps. Although the employees responsible for these activities are not explicitly listed, it appears that both: the IT and the business managers are involved. Additionally to these two sections, the responsibilities of the organisation also come along punctually in other sections or controls of annex A, but it is nowhere clarified which employee is responsible for which activity or even, which responsibilities have to be assigned to the IT or to the business staff.

Rights necessary for performing responsibility are concisely listed mostly in section 5.2, but without really being the focus of interest or a requirement for the implementation stage. The rights/capabilities listed in this section remain generic and do not deliver any necessary material for using the standard in practice.

Although the standard has not for duty to fix the responsibilities for each of these controls, it appears that the control A.6.1.3 requires that *all information security responsibility shall be clearly defined* and that the control A.6.1.1 advocates that the management *shall actively support security within the organisation through clear direction, demonstrated commitment, explicit assignment, and acknowledgement of information security policy.* This need is important since it corroborates the statement (reviewed in ISO/IEC 38500 and in COBIT) that there must be a responsibility, assigned to the management, to allocate specific and detailed responsibilities to the employees regarding each of these controls.

3.3.2 Corporate governance framework

The objective of this section is to explore how the concept of responsibility is exploited as a function of the corporate governance which impacts the access rights management. Therefore Basel II and SOX have been analysed.

3.3.2.1 Basel II

Basel2 (2004) is an international standard that aims at protecting the financial system by ensuring that banks have adequate capital to face their operational risks. The framework consists of recommendations which are issued by the Basel Committee on Banking Supervision and that banking regulators can use when creating regulations to enforce the international banking system stability, to improve the solidity of the international banks, to improve the practices of risk management, and to enhance the alignment of the capital that the banks need to put aside to guard against the types of financial and operational risks that the banks face. Basel II is elaborated upon three pillars (Figure 3.6), namely: (1) *Minimum Capital Requirements* which requests the maintenance of regulatory capital to face the credit, operational and market risk, (2) *Regulatory Supervision* that aims at examining and validating the method exploited by (1), and (3) *Market Discipline* which concerns the establishment of transparency rules related to the

information which must be publicly available. The requirements to perform the goals of these three pillars can be only fulfilled if banks have an appropriate information system. Therefore, governance of these systems plays a significant role.

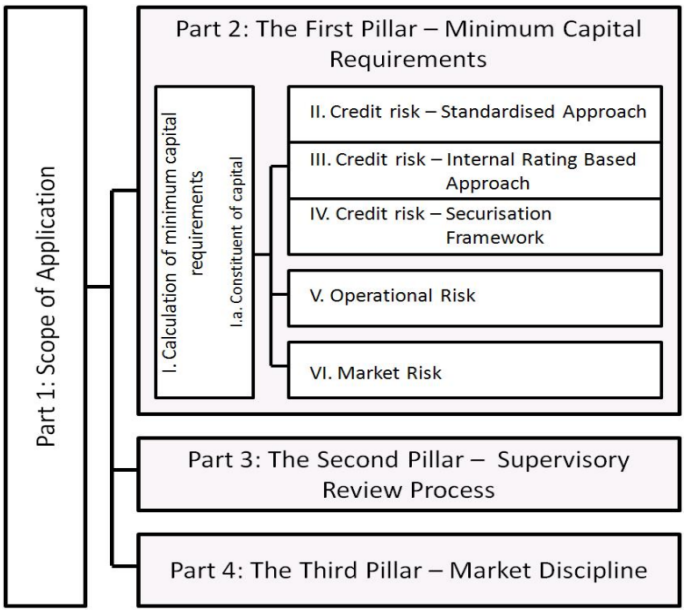


Figure 3.6: Three pillars of Basel II, **Adapted from:** Basel2 (2004)

3.3.2.2 Governance needs related to the access rights in Basel II

The following examples illustrate how Basel II addresses the governance's needs related to the definition of the responsibilities of some business roles:

- Banks must have independent credit risk control units that are responsible for the design or selection, implementation and performance of their internal rating systems. The unit(s) must be functionally independent from the personnel and management functions responsible for originating exposures.
- Supervisors should review and evaluate banks' internal capital adequacy assessments and strategies, as well as their ability to monitor and ensure their compliance with regulatory capital ratios. Supervisors should take appropriate supervisory actions if they are not satisfied with the results of this process.
- A credit risk control unit must actively participate in the development, selection, implementation and validation of rating models. It must assume oversight and supervision responsibilities for any models used in the rating process, and ultimate responsibility for the ongoing review and alterations to rating models.
- The bank's board of directors has the responsibility to ensure that management establishes a system for assessing the various risks, develops a system to relate risk to the bank's

capital level, and establishes a method for monitoring compliance with internal policies. The board should regularly verify whether or not its system of internal controls is adequate to ensure well–ordered and prudent conduct of business.

These examples highlight that Basel II addresses **two types of responsibilities**. The first type concerns the responsibilities of the business roles (e.g., board of directors, supervisors, and so forth.) which relate the business actions to be performed, as addressed in Feltus and Rifaut (2007). Secondly, Basel II also stresses the obligations of the management which is explicitly responsible for assigning responsibilities to its subalterns. This responsibility (illustrated by the last example) correlates the need to have a responsibility to generate responsibility. The latter has already been observed in the control A.6.1.1 from ISO/IEC 27000, in COBIT and in ISO/IEC 38500. This need contributes to create a "chain of responsibilities".

Basel II additionally contributes to define the specific rights required for these roles. For instance, the minimum requirement to receive recognition for eligible financial collateral implies that:

- *The legal mechanism by which collateral is given must be robust and ensure that the lender has clear rights over the proceeds from the collateral.*

- *Bank must have the rights and expectation to receive payment from the credit protection provider without having to take legal action to pursue the counter–party for payment.*

For a credit derivative[1] contract to be recognised, another requirement is, for instance:

- *The identity of the parties responsible for determining whether a credit event has occurred must be clearly defined. This determination must not be the sole responsibility of the protection seller. The protection buyer must have the rights/ability to inform the protection provider of the occurrence of a credit event.*

Finally, additional translations of the governance's needs related to the definition of the responsibilities can be retrieved in the third pillar of the framework under the form of:

- Obligations related to the responsibility of the supervisors: *Supervisors must take care to carry out their obligations in a transparent and accountable manner. Supervisors should make publicly available the criteria to be used in the review of banks' internal capital assessments.*

- The specification of the authority of the supervisors: *Alternatively, supervisors have the authority to require the banks to provide the information in the regulatory reports and the parties have the authority to approve exceptions, frequency of rating reviews, and management oversight of the rating process.*

- Type of obligation to control: *Management should ensure that the appropriate verification of the information takes place.*

[1] Instruments or techniques to separate and transfer the credit risk

3.3.2.3 Sarbanes–Oxley Act

The scope of Sarbanes–Oxley Act (SOX (2002)) is the financial reporting. This law, also named "Public Company Accounting Reform and Investor Protection Act", has been voted on July the 30th 2002 after a number of corporate scandals such as Enron, Worldcom and Tyco International. SOX describes requirements and specific mandates for financial reporting with the main objective to enforce public confidence in the markets. It is focused on the regulation of corporate governance and financial practices and aims at creating a more responsible environment for the disclosure of financial information. Three main principles guide the law: the accuracy and the accessibility of the information, the manager's responsibilities, and the independence of the audit party. The Sarbanes–Oxley Act itself is organised into eleven titles:

- Title I–Public company accounting oversight,
- Title II–Auditor independence,
- Title III–Corporate responsibility,
- Title IV–Enhanced financial disclosures,
- Title V–Analyst conflicts of interest,
- Title VI–Commission resources and authority,
- Title VII–Studies and reports,
- Title VIII–Corporate and criminal fraud accountability,
- Title IX–White–collar crime penalty enhancements,
- Title X–Corporate tax returns,
- Title XI–Corporate fraud and accountability.

The SOX introduces the establishment of a Public Company Accounting Oversight Board (PCAOB) *to oversee the audit of public companies that are subject to the securities laws, and related matters, to protect the interests of investors and further the public interest in the preparation of informative, accurate, and independent audit reports for companies the securities of which are sold to, and held by and for, public investors.*

3.3.2.4 Governance needs related to the access rights in Sarbanes–Oxley Act

The SOX is a very high level framework for the corporate governance and financials. The technical aspect of the access rights management is not the principal objective of the framework. At the business layer, the law only requests companies to provide access rights to corporate financial information to the Securities and Exchange Commission[1]. This request is mainly developed in the *Title IV–Enhanced financial disclosures.*

The law realises the governance needs related to the definition of the parties' responsibility in title III entitled *Title III–Corporate responsibility* and through the responsibility of the PCAOB developed in the *Title I–Public company accounting oversight board.*

The law, firstly, provides clearly the obligation of the responsibilities of the principal executive officer or officers. It also determines the responsibilities of the principal financial officer or officers related to the certification of each of the annual or quarterly reports. These obligations are, for instance, the establishment and maintenance of an internal control. It also encompasses

[1] http://www.sec.org

the obligation to disclose all *significant deficiencies* or *any fraud* to the audit committee. This last type of obligation is an obligation of transparency. On the PCAOD level, the law defines that, for instance, the PCAOD's duties are to conduct inspections of registered public accounting firms.

The law also stresses the importance of providing accurate information for the different stakeholders: (i) *Approval by an audit committee of an issuer under this subsection of a non–audit service to be performed by the auditor of the issuer shall be disclosed to investors*, (ii) *the Commission shall submit a report to the President, the Committee on Banking, Housing, and Urban Affairs of the Senate, and the Committee on Financial Services of the House of Representatives, setting forth,* e.g., *any recommendations of the Commission for improving the transparency and quality of reporting off–balance sheet transactions in the financial statements and disclosures required to be filed by an issuer with the Commission* (iii) *the signing officer* (e.g., the principal executive officers) *have designed such internal control to ensure that material information relating to the issuer and its consolidated subsidiaries is made known to such officers by others within.*

In this last example (iii), it is the responsibility of the officer to have access to the information although in example (i) and (ii), it is someone else's responsibility to provide information to the responsible stakeholder.

SOX finally addresses at a very high level the concepts of penalty and sanction. In section 105, *INVESTIGATIONS AND DISCIPLINARY PROCEEDINGS*: *The Board shall establish, by rule, subject to the requirements of this section, fair procedures for the investigation and disciplining of registered public accounting firms and associated persons of such firms.* [...] *If a registered public accounting firm or any associated person thereof refuses to testify, produce documents, or otherwise cooperate with the Board in connection with an investigation under this section, the Board may: (i) suspend or bar such person from being associated with a registered public accounting firm, or require the registered public accounting firm to end such association; (ii) suspend or revoke the registration of the public accounting firm; and (iii) invoke lesser sanctions as the Board considers appropriate, and as specified by rule of the Board.*

3.3.3 Summary of the governance needs related to the access rights management

From this analysis of the frameworks and definitions, we acknowledge that the general and basic need for the access rights management is that these rights must be strictly assigned to the employee who needs it for achieving a task. This requirement is justified by SOX (2002), IT Governance Institute (2007) and ISO27000 (2012) which require that the employee **responsible** for a task should possess the necessary resources to achieve this task. The standard ISO/IEC 38500 (ISO38500 (2008)) additionally strengthens this notion of responsibility through its first principle *Responsibility – employees understand and accept their responsibilities* (section 3.3.1.3), Sarbanes–Oxley Act (SOX (2002)) refers to the manager responsibilities and the responsibility of the company in the *Title III – Corporate responsibility* and Basel2 (2004) highlights and stresses the importance of the responsibilities mainly of the bank supervisors. In addition, Basel2 (2004), ISO38500 (2008) and ISO27000 (2012) emphasis the need to have one particular responsibility dedicated to the definition and assignment of other responsibilities to the appropriate employees.

The concept of resources introduced by COBIT concerns the access rights but it is however

more largely used and includes additional meanings like the employee's capabilities or other types of rights provided by the company like the decision rights (MIT (2002)) or the competences. Equally, ISO/IEC 38500 (ISO38500 (2008)) requests the employees to have the necessary competences to fulfil the responsibilities they are assigned to.

The review of the standards and norms advocates that being responsible for a task also constraints the employee to be accountable for the achievement of this task and about the use of the information system (Basel2 (2004) and ISO27000 (2012)). IFC (2002), OECD (2004), Webb et al. (2006), IT Governance Institute (2007) and ISO38500 (2008) address this need through the notion of a controlled use of the IS. The MIT (2002) extends this need and calls for an accountability framework, Webb et al. (2006) request the definition and maintenance of an effective IT control and accountability, and COBIT clearly defines, through the RACI chart, the business roles' responsibilities and accountabilities for the activities. Employees and especially directors' accountability is an important requirement for ISO38500 (2008) and Sarbanes–Oxley Act (SOX (2002)) provides clearly the obligations (including the obligation of answerability) for the main officers. ISO27000 (2012) considers that an employee is sanction–able.

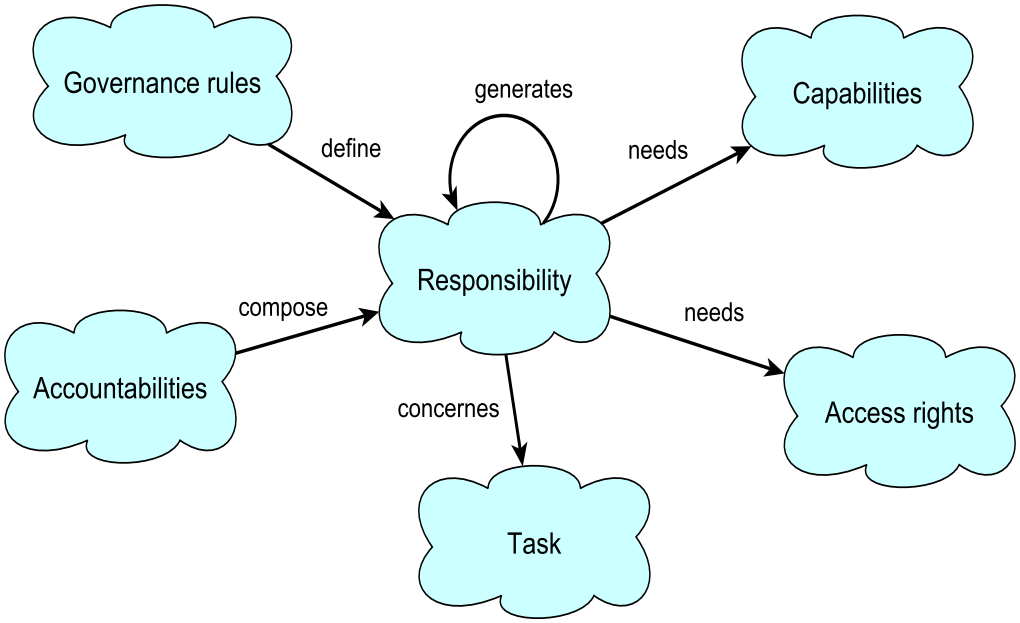


Figure 3.7: Zones of concepts

The assignment of an employee to a task and by consequence the provisioning of the rights to the employees is a process also addressed by the frameworks and the norms. The access rights assignment process is clearly defined as a necessity in IT Governance Institute (2007) which requires access rights to be requested by the user manager, to be approved by the system owners and to be implemented by the security–responsible officer and in ISO38500 (2008) which considers highly important that the employees understand and accept their responsibilities during the assignment process. Equally, ISO27000 (2012) requires an employee to be aware

of its responsibility regarding a task and IT Governance Institute (2007) requests the employee assigned to responsibility to be uniquely identified.

This analysis of the governance frameworks and definitions allow drawing the sketchy picture including significant zones of meaningful concepts, presented in Figure 3.7.

Concretely, the needs of governance for the management of the access rights which we have extracted from the analysed frameworks have been summarised in Table 3.1.

Element	**COBIT**	**ISO/IEC 27000**	**ISO/IEC 38500**	**Basel II**	**SOX**
Responsibility defined by Governance rules	X		X	X	X
Responsibility generates responsibility	X	X	X	X	
Responsibility is composed of accountabilities	X	X	X	X	X
Responsibility concerns tasks	X	X	X	X	X
Responsibility needs capabilities	X		X	X	X
Responsibility needs access rights	X	X			X

Table 3.1: Governance needs summary

3.4 Governance needs fulfilment by the access control models and rights engineering methods

Based on the review of the state of the art in Chapter 2, we analyse in this section the fulfilment of the governance needs through the reviewed access control models and rights engineering methods.

We have observed that all governance frameworks require the assignment of the rights necessary to achieve a task to be effectively realised in alignment with the business needs. This alignment is differently satisfied by the access control models. The analysis of the state of the art has permitted to understand more deeply that the models and methods reviewed most of the time consider the concept of user which is the subject that gains or not the access rights over an information and the concept of object that corresponds to the information which is accessed or not by the subject. Both concepts (subject and object) are additionally associated, depending on the models, with a set of concepts that realises the link between them. The concepts from the application layer are used with profusion in this perspective and one observes that the models do not really consider business specifications when the access rights are engineered. The Bell–LaPadula and Biba models, for instance, consider that the rights provided to an employee can be restricted, respectively, according to confidentiality and to integrity. This restriction is performed based on the subject access class (Section 2.2.1). The discretionary access control model tends to use the concept of group to gather employees with the same operations to perform on the information system. The problem, in this case, is that this concept of group has no correspondence with the employee's functions or with their hierarchic positions. The mapping between that concept and a set of employees having the same tasks to perform is, by the way, hazardous. RBAC model defines a role as a job function with some associated semantics regarding the authority and the responsibility conferred to the users assigned to it, but it does not define the responsibility and thereby, does not contribute much in the engineering of the rights to be assigned to the role owner. The ABAC model appears to be designed more to solve problems, like the mobile computing or the management of dynamic situations, then to complete RBAC regarding its mapping with business artefacts. UCON is a generic metamodel able to model RBAC and ABAC advantages at the same time and by consequence, is once again enriched at the application layer but, without improvements in terms of alignment with the business one. The solutions based on the concept of task like TRBAC or ARMF from Crook et al. are closer to the business layer since the task is a central element in the solution and is derived from business artefacts, and OrBAC puts forward the organisation to define the security rules. However, the solution existing up to date provide access rights which are engineered based on the tasks, but not on the real responsibility of the employees regarding these tasks.

Finally, the needs to have the rights provided to the employees based on their responsibility are not addressed. Although we observe a progression of the access control models and rights engineering methods towards a wider consideration of the business layer e.g., progressive introduction of the task, the group, the role (which is defined in terms of responsibility (Figure 2.2.3)), we also observe that a formal definition of this responsibility and of the accountability, as requested by the governance needs and highlighted in this chapter, has never been concretely realised before.

The next section reviews these concepts of responsibility and accountability from the litera-

ture in order to understand their meaning and how they are perceived.

3.5 Fundamentals of Responsibility and Accountability

The review of the governance's needs has highlighted the importance of having the responsibility and accountability suitably defined, assigned with the strictly necessary rights and capabilities, held by employees which are accountable regarding the task they have to perform.

As the objective of the research is to improve the definition of the access rights considering these needs, in the sequel of this chapter, we review their signification and how they have already been addressed in the literature. An initial observation we do starting this review is that the literature corresponding to the responsibility and the accountability is not only focused in one field but is spread in many disciplines. As the objective of the research is not to complete this state of the art but rather, to apprehend it to rightly consider it in the sequel of the research, the review is limited to the most significant papers related to each concepts.

3.5.1 The concept of Responsibility in general

The word *responsibility* is a common word often used. The responsibility has many meanings, e.g., the state assigned to a person which is answerable for actions or for those of someone under its responsibility, the person which is guaranteeing something, the person who is the cause of a mistake, or who has a role to play and the power to make decisions. From a scientific point of view, the responsibility has been subject to many investigations in many science disciplines and each discipline has elaborated its own theory based on its own perception of it. In the field of economics, Sliwka (2006) considers that the responsibility exists when *a superior holds a certain subordinate responsible for a task, when he announces his beliefs that this subordinate contributes most to this task*, although at an organisational point of view, Prendergast (1995) considers that *the responsibility of an agent is defined as the subset of tasks allocated to him by a manager and it is shown that rent seeking considerations lead the manager to allocate the few tasks to the agent*. In this section we review this concept to understand its meaning and the signification of the concepts that compose it. The review is performed by analysing how it is perceived through various disciplines.

To review the concept of responsibility, we introduce it by analysing a paper from Vincent (2011). This paper presents a structured taxonomy of the responsibility concepts (STRC) from the philosopher Nicole Vincent. This author considers that the responsibility is a "syndrome" of concept[1]. Beginning with the review of this paper is interesting because it permits to introduce the responsibility through six perspectives that describe and propose well determined meanings of this concept.

1. The first meaning considers the *virtue responsibility*, that reflects the character or reputation of a person who takes his/her duties seriously. It reflects by the way, following Vincent (2011), the commitment of the person to do what he supposes to be right.

[1]For the author a "syndrome" of concept is when multiple concepts are shared by a common word.

2. The second perception of the responsibility is the *role responsibility* that reflects what a person should or should not do because of a conventional, an institutional, or a social role. Nicole Vincent considers the role responsibility as the duty to be performed by the person.

3. The third perception introduces the *outcome responsibility*. It represents the consequence of something for which a person (or an entity) is imputable (to something that he/she performs or not). This is the usual perception of the philosophers.

4. The fourth perception of the responsibility is the *causal responsibility*. In this sense, the responsibility reflects a person or an object that has caused something. This is different from the outcomes responsibility since, for Vincent (2011), it has a less morally implication.

5. The fifth perception of the responsibility is the *capacity* that reflects the mental capacity of a person. This is a more psychological perception. Children and persons mentally ill are, e.g., not to be considered as responsible in that sense.

6. The sixth perception depicted is the *liability* responsibility that points out who takes the responsibility for what happens.

Nicole Vincent also reviews the relation between the six senses of responsibility that is represented in Figure 3.8, extracted from Vincent (2011).

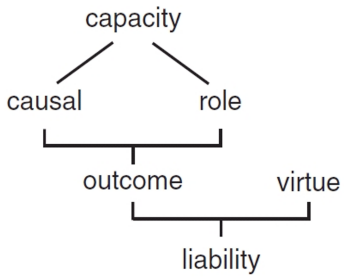


Figure 3.8: Six senses of responsibility, **Source:** Vincent (2011)

Mainly three types of relations bind the six different senses of the responsibility. We illustrate them through some examples from the healthcare domain:

1. The first relation is Outcome Responsibility from Causal and Role responsibility.
   To illustrate the relation, let us take the example of a nurse in a hospital who makes an injection of penicillin to patient that dies a short time after the injection because he was allergic to the penicillin. The nurse is outcome responsible for the death of the patient because she had to verify if an allergy existed. In this case, the nurse violated her role responsibility which is to verify allergies before performing an injection.
   Suppose now that the nurse asked the patient about the allergy and that the patient hides his allergy. In this case, the patient is outcome responsible for its own death although the nurse keeps the causal responsibility of it.

2. The second relation is the Capacity Responsibility to Causal and Role Responsibility.
   In the above case of the penicillin injection, let us suppose now that the nurse performs the injection on Monday morning and that she was on duty, at the hospital, since Friday

night. She was consequently extremely tired and she forgets to ask the patient about its allergies. In this case, she lacked the capacity to remember to ask about the allergy. It could also happen that the patient did not remember about its allergies because he was chocked after a crash. This example illustrates that it is important to analyse the capacity of the stakeholders to determine if they can be considered mentally responsible to their acts.

3. The last relation is the liability from outcome and virtue responsibility
   The person that is outcome responsible, the nurse or the patient, is also the person that in principle has the liability responsibility. The virtue responsibility is however sometimes considered to mitigate or aggravate the sanction. For instance, if the nurse is recognised as a very good subject who realises, most of the time, her duties seriously, the liability responsibility can be mitigated.

3.5.2 The concept of Accountability

Accountability covers one aspect of the responsibility (Mulgan (2000)). Accountability in the sense we use it nowadays, such as the concept of governance, did only appear some decades ago (Mulgan (2000)). The commonly agreed meaning is to give account to someone who has authority for actions. However, what we observe during the review of the Accountability concept in the literature is that it can be apprehended through many perspectives (Day and Klein (1987), Sinclair (1995), Erkkilä (2007), Bovens (2010), Dubnick and Yang (2010) and Blind (2011)) and that there exists an abundance of definitions: Romzek and Dubnick (1987) defines it in the context of a public administration as *the means by which public agencies and their workers manage the diverse expectations generated within and outside the organisation.* Day and Klein (1987) explains that it is *all about the construction of an agreed language or currency of discourse about conduct and performance* and also that it includes *the criteria that should be used to assess it.* Day and Klein (1987) additionally argue that accountability encompasses the social consensus that argues for good conduct and acceptable performance. Mulgan (2000) defines it as *the process of being called to account to some authority for one's action.*

The origins of accountability date back to the 11^{th} century, in England, where the *Domesday Book* was established to make inventory of all the possessions of the king. The sense of accountability at that time was twofold. The first concerns the obligations to report what really exists in the realm, the second aims at knowing and judging the conduct of the realm's property holders. During this period, the main concern of the accountability was the possessions (what the property holders had) and the secondary concern was the justification of their behaviour (what the property holders did).

Nowadays, those two perspectives still exist. If one looks forward in the Bovens' research (Bovens (2010)), one observes two types of accountability concepts. The first type is the accountability as a virtue and the second type is the accountability as a mechanism.

The accountability as a virtue corresponds to a type of communication tool that provides, mostly in the field of politics, a sense of good governance (Dubnick (2007)). It strives to convey images such as transparency, liability, dialogue, participation or involvement. Mulgan (2000) has defined the virtue responsibility as *a general term for any mechanism that makes powerful institutions responsive to their particular public.* Bovens (2010) explains that being accountable

is seen as a virtue, as a positive quality of organisations or officers. Both Bovens (2007) and Dubnick (2007) complete their definitions with the notion of being accountable for the entity that holds the accountability, *by a forum.*

Structuring the virtue of accountability in standards and norms is infrequent and challenging due to the loose nature of its apprehension. In the same way, it is also difficult for an organisation to set up a structured framework for its control (Bovens (2007)). M. Blagescu and Lloyd (2011) provide the Global Accountability Framework that includes four dimensions:

- Transparency,
- Participation,
- Evaluation,
- Complaint and response mechanisms.

The other type of accountability concept refers to the accountability as a social mechanism. This accountability is defined by Bovens (2007) as *a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can ask questions and pass judgement, and the actor may face consequences.* In this definition, the first statement says that the accountable actor is obliged to inform the forum about the performance of tasks. By extension, the information provided also concerns the achievement of a procedure or the outcomes.

The second statement argues that the forum can ask questions and pass judgement. Although the first statement is an obligation of the accountable actor, the second statement is a task to be handled by the forum.

The third statement provoked by the previous one reflects the necessity for the accountable actor to face the consequences resulting in the appreciation of the achievement of its obligations. Facing the consequences most of the time takes the form of positive or negative sanction. Mulgan (2000) also considers the sanction as constituting the accountability. Positive sanctions are for instance a reward, a recognition, a payment of money and negative sanctions can be a disciplinary measure, a civil remedy (in the case of a political act) or a penal sanction.

In Blind (2011), the author highlights, through different papers, the dichotomies observed at a scholarly level related to this concept. For Blind, the accountability can be defined as *an abstract and value–ridden concept* or as a *highly concrete and value–free concept* where origins lie in bookkeeping. The first definition refers to as the state of *being answerable* although the second one *to the obligation to give evidences of management or performance, imposed by law, agreement or regulation.* He explains that the observed dichotomies can be classified along four perspectives: prescriptive, descriptive, operational and longitudinal. Each of these four perspectives offers a way to understand the dichotomies. The prescriptive dichotomy concerned the virtual level of the accountability. The Bovens (Bovens (2010)) dichotomy explained here is a typical case of it. One sense of accountability reflects the philosophy and the other reflects the means (Blind (2011)). This is respectively, according to Bovens, the virtue accountability and the accountability as a social mechanism. The descriptive dichotomies are complementary to the prescriptive dichotomies and refer to the domains of perception. An example of descriptive dichotomy is the Market–based versus the administrative based accountability. This depends if the justification of the accountability is towards the market where formal rules are not really

existent or towards an administration using formal rules. The operational dichotomies reflect if the accountabilities are horizontal or vertical. That signifies, if the accountability is held to the peers or external evaluators. The longitudinal dichotomy of accountability explains if the accountability concerns past act or if it concerns "responsibility" of the accountability holders like the expected responsiveness of a government to the citizens (Blind (2011)).

Many authors (Erkkilä (2007)) present the accountability mainly through four types: the *Political accountability* that concerns the people elected who have to answer for their acts to the public, the *Bureaucratic accountability* that reflects the obligation for a subordinate to be accountable towards a superior due to the hierarchic relationship between them, the *Personal accountability* that expresses the feeling of internal control of a person and by the way its personal integrity, finally the *Professional accountability* that represents the sense of duty felt, due to the membership of a professional group. This sense of accountability is very close to the understanding of the value commitment despite that it is not sharing the idea that the employee possesses the same value as the organisation (O'Reilly and Chatman (1986)). A fifth type of accountability introduced by Sinclair (1995) is the *Managerial accountability*. This *Managerial accountability* appears to be similar to the *Administrative accountability* but is however slightly different since the first type refers to monitoring the input and output of a process although the last type mostly concerns the monitoring of the process itself.

The concept of accountability is composed of sanction (Mulgan (2000)). At the origin, the concept of sanction meant the enactment of a law, or the confirmation by an authority of something considered as solemn or holy. The review of its meaning nowadays shows that it tends to evolve towards the consequences resulting in the evaluation of tasks performance. For Bovens (Bovens (2010)), an actor is formally or informally implied and subject to sanction in case of bad performance or to a reward in case of adequate performance. For him, the sanction is a constitutive element of the type of *accountability as a mechanism* and can be a *rather formal and legal connotation*. Bivins (2006) explains that some actors can be motivated by the sanction that he calls rewards or punishment. Dubnick considers that the threat of sanction, even if it is not used as an active tool, can act as a background reminder for the actor about its moral obligation (Dubnick (2007)). In Fox (2007), Fox considers that the accountability may be *soft* or *hard* depending if it includes sanction or not.

3.5.3 The concept of Responsibility in IT

This section aims at providing a view on the perception of the concept of responsibility in the IT domain. To date, this concept has been poorly addressed by the research concerned in the management of IT and authors having published on those topics are limited. Storer and Lock (2008) define the responsibility as *duties which are to be discharged by agents*. Sommerville et al. (2009b) complete this definition and precise that the duties exist in order *to achieve, maintain or avoid some given state, subject to conformance with organisational, social and cultural norms* and Stahl (2006) introduces the notion of answerability: *The responsibility is the ascription of an object to a subject rendering the subject answerable for the object.*

Martin et al. (2005) present an interesting work to introduce the multi–facet of the responsibility in IT. In this paper, the authors collected a set of problems that could occur around the responsibility of a team of analysts and developers engaged in the development and the deployment of an electronic patient's records system (ERPs) in the National Health Service

(NHS) in England. Through some well defined examples, they illustrated the sharing of responsibilities between the designer and the users of the system. The paper presents, e.g., how a responsibility can be transferred from the system user to the designer when the information, necessary for the design, provided by the user on the processes is not accurate, and so forth. Martin et al. analyse the work of the designers and their relationships with the users under the ethnographic perspectives and they try to bring solutions regarding the assignment of the responsibilities in the every–day design work. In Sommerville (2007a), Sommerville also lists six types of responsibility vulnerabilities to introduce his work: (1) unassigned, (2) duplicated, (3) uncommunicated, (4) misassigned,(5) responsibilities overload and (6) responsibility fragility.

Strens and Dobson (1993) address the responsibility concept to consider the security of the information system and they advocate that the security must be perceived through a socio-technical approach rather than only through a technical point of view. Without defining a formal model of responsibility, they explain that the responsibility is built around three types of needs: the need to know, the need to do, and the need to show how the responsibilities are fulfilled. Based on the responsibility, they explain that the designer of a system can better understand what the user needs. For the author, the obligations are linked to the agent that performs activities and by doing these activities, the agent fulfils his/her obligation.

Strens and Dobson (1993) define the responsibility in the perspective of a relationship between agents. One of the agents gives the responsibility whereas the other agent receives it. In the frame of a delegation, based on this perspective, when a responsibility is transferred from one agent to another, a new responsibility is created and both agents are assigned to a type of responsibility and to a type of obligation. The agent that transfers the responsibility is called the responsibility principal and the agent that receives it is called the responsibility holder.

Strens' point of view, about the delegation of the responsibility, is that even if the responsibility principal does not perform the task that he has delegated to the responsibility holder, he remains answerable for that task. Although we could agree with the fact that the principal keeps his/her responsibility when the delegation happens between a manager and one of its subordinate, it is disputable when the transfer of responsibility happens between two agents at the same hierarchical layer. For instance, if a nurse accepts to replace one of her colleague during the pause, this colleague is responsible for the patients during this pause and she is answerable for her acts.

Cholvy et al. (1997) are interested in formally modelling the concept of responsibility in the field of IT. For the authors, this formalisation is complex due to the different meanings of the responsibility. In the paper, Cholvy et al. review three meanings of the concept and explain how formalising the responsibility is fundamental to improve the behaviours of the systems and the organisations. She depicts the three following definitions:

1. Something bad happened and you could have prevented it.
2. Obligation or moral duty to report or explain your actions or someone else's action to a given authority (answerability).
3. Position, which enables you to make decisions in a given organisation but implies that you must be prepared to justify your actions (accountability).

Those three definitions will be reviewed later in this section for the elaboration of the Responsibility model.

3. NEEDS OF GOVERNANCE AND FUNDAMENTALS OF RESPONSIBILITY

Ian Sommerville in Sommerville (2007a) introduces the Responsibility Assignment Models that have for objective to facilitate the distribution of responsibilities in a system. For Sommerville, there is flexibility in the assignment of responsibility that may always be subject to negotiation. This negotiation takes place during the design of the socio–technical system and permits to set up hyphens between automated and manual tasks.

Sommerville defines two types of responsibilities: causal and consequential. Consequential responsibility reflects who gets the blame or credit for the occurrence of some state of affairs (this perception of the responsibility corresponds to the liability responsibility in Vincent (2011)) whereas causal responsibility reflects who or what is responsible for making something happen or avoiding some undesirable state (this perception of the responsibility corresponds to the role responsibility in Vincent (2011)). Sommerville expresses that the Authority is also a significant concept linked to the responsibility and that, once a responsibility is assigned to someone, there should be an authority which decides whether or not the responsibility has been properly discharged. According to Sommerville, in some cases, the responsibility can be shared between multiple agents. Three types of responsibilities have been depicted: the joint responsibility, the derived responsibility, and the delegated responsibility.

In Sommerville (2007b), Sommerville proposes a model of the causal responsibility. As introduction, he depicts the advantages of modelling the responsibility without considering the agent that will be assigned to this responsibility. The four advantages are: (i) it focusses on the responsibility itself and on the intention of the organisation, (ii) it permits to analyse the relationship between responsibilities, (iii) it provides a basis for the assignment of responsibilities and (iv) it provides a basis for vulnerability analysis (i.e., do the agents have the requested capabilities, competencies, resources, and so forth).

Sommerville introduces a granularity in the responsibility that he considers to be simple or composite. A composite responsibility is made up of simpler responsibilities that are coherent and mutually dependent. The simple responsibility corresponds to our understanding of the responsibility. Sommerville distinguishes three classes of causal responsibility *Doing*, *Monitoring* or *Avoiding*. Its proposed pattern for responsibility description includes the following concepts: *Name, Context, Type, Classification, Pre–conditions, Post–conditions, Normal process, Variations, Exceptions, Advice* and *Requirements*. Moreover, the author considers that one role may be composed of responsibilities. In Dobson et al. (2006), the authors exploit the definitions of causal and consequential responsibilities and focus on this concept of responsibility for designing information systems. The authors argue that using the responsibility permits to better analyse the problems that could arise during the design since it permits to depict organisational failure.

In Sommerville et al. (2009a), the author refines the requirement engineering process by enhancing the question *What should the system do?* by *What do the stakeholders need and produce?* He advocates that the modelling of the responsibility can contribute to that enhancement and, therefore, proposes an approach based on three levels:

1. The analysis of the documents.
2. The interviews of the stakeholders.
3. The field observations.

The finality of it is the discovery of information requirements in order for the stakeholder to discharge the responsibility and the translation of these information requirements into the system requirements.

The major difference between Sommerville's work and our work is the relation between the concept of accountability and the concept of responsibility. For Sommerville, only the consequential responsibility implies accountabilities and refers toward an authority that is another agent (person or organisation). The causal responsibility does not imply accountable. In our point of view, each agent is accountable, should it be the agent that does the procedure of a task or the one that is accountable for the achievement of its goal.

To represent the responsibility, a graphical model has been introduced by Blyth et al. in ORDIT (Blyth et al. (1993)). The ORDIT methodology aims at representing human resources and the technological system to achieve the organisational goals. Sommerville has proposed a complementary graphical notation that he illustrates in Sommerville et al. (2009a) and Sommerville et al. (2009b).

3.6 Conclusions

In Chapter 2, we have observed, through the analysis of the state of the art, that the evolution of the solutions proposed tends to consider more and more the concepts from the business layer as significant for the management of the access rights. Therefore, the concepts exploited evolved toward more business oriented concepts such as the task to be achieved, the group to which the user belongs, or contextual information such as the location of the user or the time of access. In the same manner, the concept of role from the RBAC model has been defined based on the responsibility, but the latter has not been thoroughly nor formally modelled in the information system field.

In this chapter, we have firstly reviewed the definitions of the corporate governance, the IT governance and the business/IT alignment. Afterwards, we have described five standards to explore how, at the governance level, they require to address the business concepts and the responsibility for the management of the access rights. This review of the governance frameworks has confirmed the importance of considering these latter and has allowed to draw an unrefined picture of zones of concepts to be handled, according to the governance needs, for this access rights management. These zones of concepts include the responsibility, the tasks, the accountabilities, the capabilities and the access rights.

Despite this progression of the access control models and rights engineering methods towards more alignment with the business layer, we have observed that the application layer and the business layer do not yet fully share a common understanding of these business concepts, responsibilities and accountabilities. For the moment there is no solution allowing connecting both layers and practically, no common model is agreed yet among the business and the IT staff. In order to fill this lack of connection between the layers and to share a common understanding of these concepts, the second part of the chapter reviews the fundamentals of responsibility, as well as the different aspects which compose it. This review does not aim at providing a rigorous state of the art but, rather an introduction to the theories that address these aspects. The latter will be used in Chapter 4 to apprehend the concepts of responsibility and accountability and to

enrich their descriptions and integrations in the Responsibility metamodel.

Chapter 4

Responsibility MetaModel (ReMMo)

4.1 Introduction

The state of the art in Chapter 2 has highlighted that the role concept, as defined in the role based access control model (Section 2.2.3), aims at gathering a set of users with the same **responsibilities** and as a result who are allocated with the same permissions. However, although RBAC considers responsibility as a centric element for defining the role, as far as our knowledge goes, this latter has not been extensively and precisely modelled in the information system field.

In practice, this lack of understanding and modelling of the responsibility leads, in some cases, to wrong interpretation of the notion of role. Indeed, we have noted in different companies, through real situations (Section 1.4) that, at the business layer, employees are assigned to business roles where the corresponding responsibilities are roughly defined, are incompletely expressed, or do not exist at all. The deficiencies in the definition of the roles result, in the field of the access rights management, in an erroneous analysis of the access rights which are, afterwards, assigned to the business role owner, and are necessary by the latter to achieve the above responsibilities.

In this chapter, we analyse in details, based on the literature presented in Chapter 3, what the concept of responsibility means and how it allows connecting the access rights and the business roles. We propose to extend existing work by creating ReMMo, a Responsibility metamodel for modelling a rich concept of responsibility, the accountabilities that are part of it and its links with the employees, the business roles, the tasks, and the rights and capabilities. Enhancing the understanding and the modelling of the responsibility contributes, on the first hand, to improve the definition of the role and, thereby, the management of the access rights, and on the second hand, to fulfil the governance needs corresponding to the responsibility as reviewed in Chapter 3.

4.2 Methodology for building the Responsibility metamodel

To elaborate the ReMMo, we analyse how the concepts that compose the responsibility are presented in the scientific literature. Practically, for each concept, we review and introduce its origin, we provide our own definition for it, we integrate it in the Responsibility metamodel, and

we associate it with other concepts. Given that ReMMo has been elaborated following an action design research method, intermediary versions of the metamodel have been designed. Appendix G presents and evaluates one of those intermediary versions of the Responsibility metamodel. This chapter presents the last version of it.

All along the chapter, to illustrate the semantics of the metamodel, we will use a simplified case study in the healthcare domain. This latter is described as follows:

In the healthcare domain, the goal is to ensure patient care. To achieve this goal, it is necessary to hire employees who mainly treat patients. In a hospital, hiring employees is a goal under the responsibility of the CEO but taken in charge by the doctor general. The treatment of the patients is a task achieved by analysing the patient's pathology and giving him drugs. To analyse the pathology, it is necessary to seek information in a knowledge base and to make X–Ray analyses. During the treatment, a report about the pathology must be prepared and the team that provides care must be supervised by one doctor. Finally, to be able to seek information in the knowledge base, employees must be instructed on how to seek this information and must therefore receive the appropriate training.

Along this chapter, we illustrate how the responsibilities are defined using our metamodel for achieving the goal and performing the tasks, how those responsibilities are assigned to the healthcare actors, and how the responsibilities are adapted according to some conditions. We also show how the rights and capabilities are provided to the employees.

4.3 Scope of the metamodel

The Responsibility metamodel aims at modelling the responsibility of actors concerning tasks. Hence, the responsibility concerning other duties are not taken into account in this metamodel, and we express the following limitation as: the Responsibility metamodel concerns exclusively the realisation of tasks, and other types of responsibilities like the *virtue responsibility*, the *causal responsibility* or the *mental capacity* (respectively the first, fourth and fifth meaning of responsibility from Vincent (2011)) are not addressed.

The elaboration of the metamodel of responsibility was guided by the objectives to model the responsibilities of the employees in a professional context and for professional purposes, and to enable an easy integration of this metamodel with other standards, norms and frameworks. The metamodel of responsibility has, accordingly, been kept as simple as possible, and without superfluous concepts.

Based on our review of the state of the art, we have noted that governance requires the performance of structural tasks (Section 3.2) in parallel to the business tasks. Based on this information, having a good understanding and a good representation of these tasks are fundamental for the modelling. Therefore, the Section 4.4 focuses on understanding and modelling the task under consideration in our metamodel.

Figure 4.1 represents the main concepts of the Responsibility metamodel. In practice, we note that tasks related concepts (represented in yellow) are assigned to a business role, or sometimes directly to employees, through the accountabilities that are part of their responsibilities

(all four in green). Although different actors (roles or employees) may be assigned responsibilities concerning the same tasks, these responsibilities have different meanings most of the time. For instance, a manager may be responsible for achieving the goal of a task although he is responsible for performing the procedure to achieve that task. The modelling part of the responsibility is presented in Section 4.5. The accountability may also exist and be impacted respectively by conditions (in green) and governance rules (in grey).

Finally, rights and capabilities (both in orange) are required in order to perform these responsibilities and accountabilities. Therefore, Section 4.6 addresses these concepts.

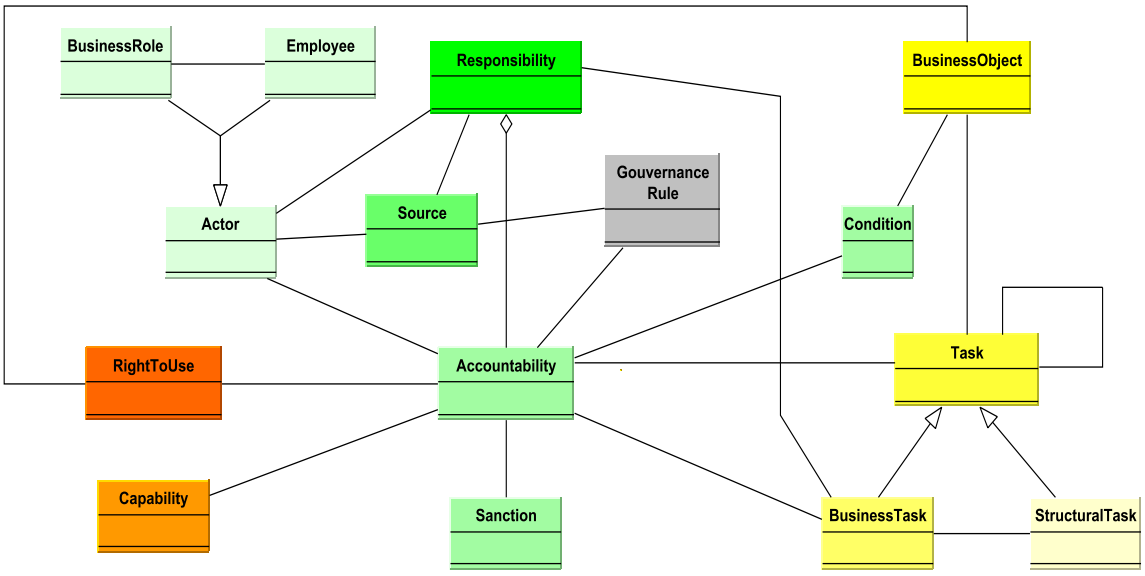


Figure 4.1: Responsibility metamodel uncluttered

4.4 Task and Business Object modelling

In this section, we introduce the concepts of business and structural task, and the concept of business object. These concepts are represented in Figure 4.2.

4.4.1 Business Task

According to Cohen (1999), *the driving force behind the creation of informing environments and delivery systems is that a task needs to be accomplished.* Based on this statement, many authors have attempted to better understand the task to be accomplished (Gill and Hicks (2006)). For Bertino et al. (2001) and White (2004), a business task is an element that composes a business process and that pursues a business goal. In that regard, Waller (1997) explains that a task is *a duty assigned to or assumed by an individual, the performance of which directly contributes to the attainment of an assigned goal* and Paterno (2001) explains that, in the field of software engineering, *tasks are activities that have to be performed to reach a goal* and he describes a

goal as *either a desired modification of the state of an application or an attempt to retrieve some information from an application.* For Paterno, tasks can also be divided into *sub–tasks of lower complexity and the relationship between the tasks can be modelled in various ways.* Hackman (1969) completes this and argues that an employee assigned to a task has a set of goals to be achieved, instructions to be performed, or a mix of both. Amyot et al. (2009) argues that tasks represent *solutions to the realisation of goals or soft goals. In order to be achieved or completed, soft goals, goals, and tasks may require resources to be available* and, in BPMN (White (2004)), a task represents a *single unit of work that is not or cannot be broken down to a further level of business process detail without diagramming the steps in a procedure.* Atluri and Warner (2005) defines the task as *a logic step or description of a piece of work that contributes toward the accomplishment of a process.* In ArchiMate 2.0, the business task corresponds to a business activity which is defined as a specialisation of a more generic business process.

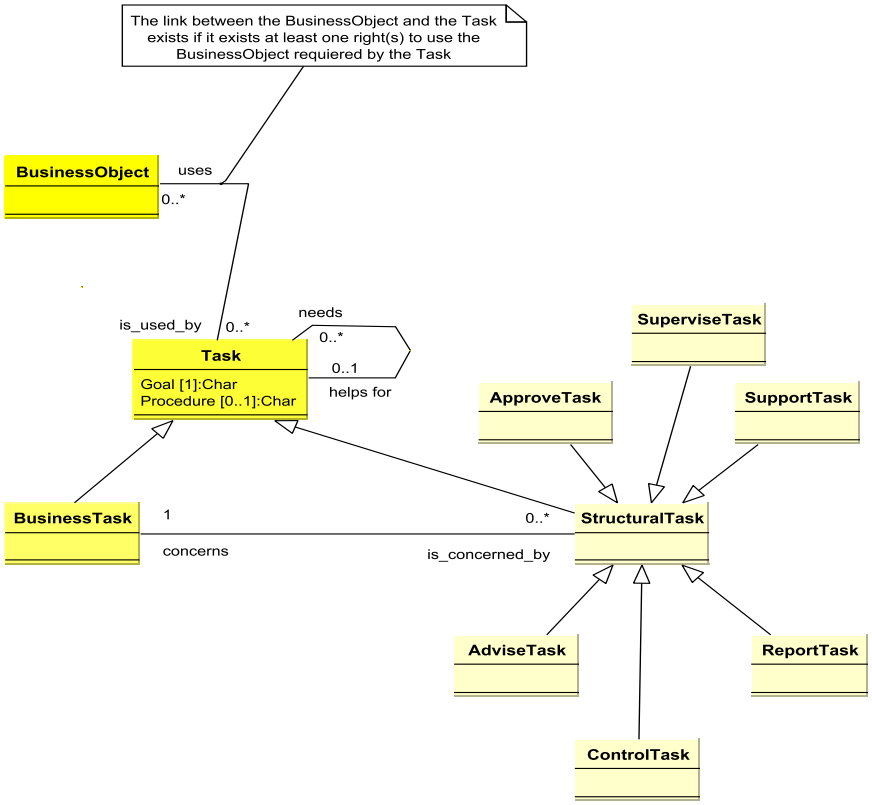


Figure 4.2: Task and business object modelling

Beside these definitions, the field of requirement engineering and, in particular, the i* framework has been elaborated to also model the concept of task according to the two above perspectives: *Goal to achieve* and *Procedure to perform.* In these models, actors depend on each other to achieve a goal, to perform a task or to furnish resources. Hence, i* expresses that an element is a task when specific ways of accomplishing it exists and is a goal when the actor beliefs that there may exist more ways to achieve it. Beforehand, Yu defined in his PhD dissertation (Yu (1996))

a goal as: *a condition or state of affairs in the world that the actor would like to achieve. It is expressed as an assertion in the representation language. How the goal is to be achieved is not specified allowing alternatives to be considered*, and a task as: *a task specifies a particular way of doing something. When a task is specified as a subcomponent of a (higher) task, this restricts the higher task to that particular course of action.* i* also defines soft goal as qualitative criteria used to evaluate alternative means for achieving ends.

In the SR diagram, elements are linked together by means–ends relationships. These links indicate a relationship between an end, and a means for attaining it. The "means" is expressed in the form of a task (since the notion of task embodies how to do something) and the "end" is expressed as a goal. Elements may also be linked together by task–decomposition relationships when a task is decomposed into sub–tasks. A task–decomposition link provides a hierarchical description of intentional elements that make up a routine. A means–end link provides understanding of the reason why an actor engages in some tasks pursues a goal, needs resources, or wants a soft goal (Yu (1997)).

4.4.2 Structural Task

The review of the governance in Section 3.2, of managerial frameworks and of real enterprise cases, highlights that, beside the business tasks, there exists a structure which supports the execution of these business tasks. This structure is composed of tasks that are called governance tasks when they contribute to achieve governance goals like the goal *to comply with regulation during the performance of business tasks*, management tasks when they contribute to management goals like *keeping the employee motivated during the performance of the business task*, administrative tasks when they contribute to administrative goals like *reporting the achievement of business goals*, security tasks when they contribute to security goals like *having the business risk under control* and so forth.

To understand the meaning of the structural task, we provide a sample of some of the most used structural tasks partially extracted from IT Governance Institute (2007): Approve, Supervise, Support, Advice, Control and Report. Table 4.4 provides a summary of the meaning of each of them.

Structural Task type	**Meaning**	**Reference**
`ApproveTask`	Is the task that aims at granting authorisation or consent for something	Blokdijk and Menken (2008)
`SuperviseTask`	Is the task that aims at directing and overseeing the performance of the team that executes something	Proctor (1994)
`SupportTask`	Is the task that aims at being present and giving help for completing something	Hightower (2008)
`AdviceTask`	Is the task that aims at providing an opinion about the best solution to choose for the realisation of something	IT Governance Institute (2007)

Structural Task type	Meaning	Reference
`ControlTask`	Is the task that aims at checking whether the acceptance criteria regarding the realisation of something has been reached	Blokdijk and Menken (2008)
`ReportTask`	Is the task that aims at providing a formal opinion and statement related to the result of a task	Soroczak and McDonald (2006)

Table 4.1: Example of types of structural tasks

4.4.3 Task

Based on review of the literature and the analysis of the meaning of the business task and of the structural task, we express that the responsibility may concern a `Goal` or a `Procedure`. Note that in the sequel of the thesis, we use the `typewriter` style when referring to concepts from the metamodel. And, according to i*, we define a task as:

DEFINITION 1: The *task* is a complete and identifiable piece of work necessary to achieve a goal and that may or may not be defined with a procedure.

Additionally, seeing that the task may be business or structural, we complete the DEFINITION 1 with:

The *task* may be either a business task if it aims at achieving a business goal or a structural which if it aims at achieving a structural goal.

The concept task is represented as a `Task` class in the Responsibility metamodel (Figure 4.2). As explained in i*, actors depend on each other to achieve a goal, to perform a task, or to furnish resources (this latter case will be addressed afterwards). In order to be compliant with the first two dependencies, while keeping the task as the unique concept concerned by the responsibility, we consider that both types of i* dependencies are `Task` types in the Responsibility metamodel.

To model this, we consider two types of attributes for the `Task`: the `Goal` and the `Procedure` and we express that one `Goal` always exists to define a `Task` although one `Procedure` may or may not exist. `Goal` and `Procedure` have thereby, respectively, the cardinality [1] and [0..1]. If a `Task` is defined by a `Goal` but no `Procedure`, then it models the "achieve a goal" in i* and if a `Task` is defined by a `Goal` and a `Procedure`, it models a "perform a task". The latter having for objective to "achieve a goal".

To illustrate this, according to the example of Yu (1997) relating to the meeting schedule, *AttendMeeting* is a goal that corresponds in our metamodel to a `Task` with the `Goal` to `AttendTheMeeting` and `EnterDateRange` is a task that corresponds to a `Task` with a `Goal` to have a data range and a `Procedure` to enter data range.

In practice, we have noted that the business is articulated with a set of business processes are decomposed into an ordinate set of tasks. This statement has been corroborated by Bertino et al. (2001) and White (2004). These tasks may afterwards be themselves decomposed into sub–task(s) (Paterno (2001)) and so forth until the last task of a chain of tasks cannot be decomposed any more. In the Responsibility metamodel, we model this by a recursive association of `Task` to `Task` such as, in the first direction, one `Task needs` zero to many `Task` and in the other direction, one `Task helps for` zero or one `Task`. In i*, this decomposition exists through means–ends relationships which associate the tasks that are performed to achieve goals, and by task–decomposition relationships which associate tasks with sub–tasks. The means–ends relation is represented, in ReMMo, by a link which formalises that a `Task` (defined with a `Goal` and a `Procedure`) `helps for` a `Task` defined with only a `Goal`, and the task–decomposition is represented by a link which formalises that a `Task` (defined with a `Goal` and a `Procedure`) `helps for` another `Task` (also defined with a `Goal` and a `Procedure`). These associations between the `Tasks` create a graph which must be of type directed acyclic graph in order to avoid going into an endless loop by always repeating the same paths of `Tasks`.

The means–ends relationships generate edges between `Tasks` such that the starting endpoint (the `Task` that `needs` another `Task`) is a `Task` defined by a `Goal`, and the ending endpoint (the `Task` that `helps for` another `Task`) is defined by a `Goal` and a `Procedure`. As we have constrained the graph of `Tasks` to be a directed acyclic graph, we achieve a partial order between the `Tasks` such that the `Tasks` only defined by a `Goal` tend to be the starting endpoints although the `Tasks` defined by a `Goal` and a `Procedure` tend to be the ending endpoints (illustrated in Figure 4.3). A parallel may be observed between the graph of `Tasks` and the organisational hierarchy (e.g., the `CEO` is the superior of the `DoctorGeneral` which is the superior of the `Doctor` which is the superior of the `Nurse`) such that the starting endpoint `Tasks` tend to be concerned by `Responsibilities` assigned to `Roles` at the top of the hierarchy and the ending endpoint `Tasks` to `Responsibilities` assigned of `Roles` at the bottom of it.

In the Responsibility metamodel, we consider that a task may be either a business task or a structural task, and we account that structural tasks aim at achieving the above structural goals. We model accordingly a `BusinessTask` class and a `StructuralTask` class which are specialisations of the `Task` class, and we express that the one `StructuralTask` concerns one `BusinessTask` and, inversely, that one `BusinessTask` is concerned by zero to many `StructuralTask`. Table 4.4 provides a summary of the main types of structural tasks. We model, in ReMMo, these types of structural tasks as classes that are subclasses of the class `StructuralTask`.

4.4.4 Business Object

A business object is either a piece of information, a document, or a material object (Becker et al. (2011)) which is, for Bruno and Torchiano (2000), produced by a business process. It enables better and more consistent binding of a real–world concept, representing a product or service which is the goal of a business activity, with the actual business process for its realisation (Katranuschkov et al. (2007)).

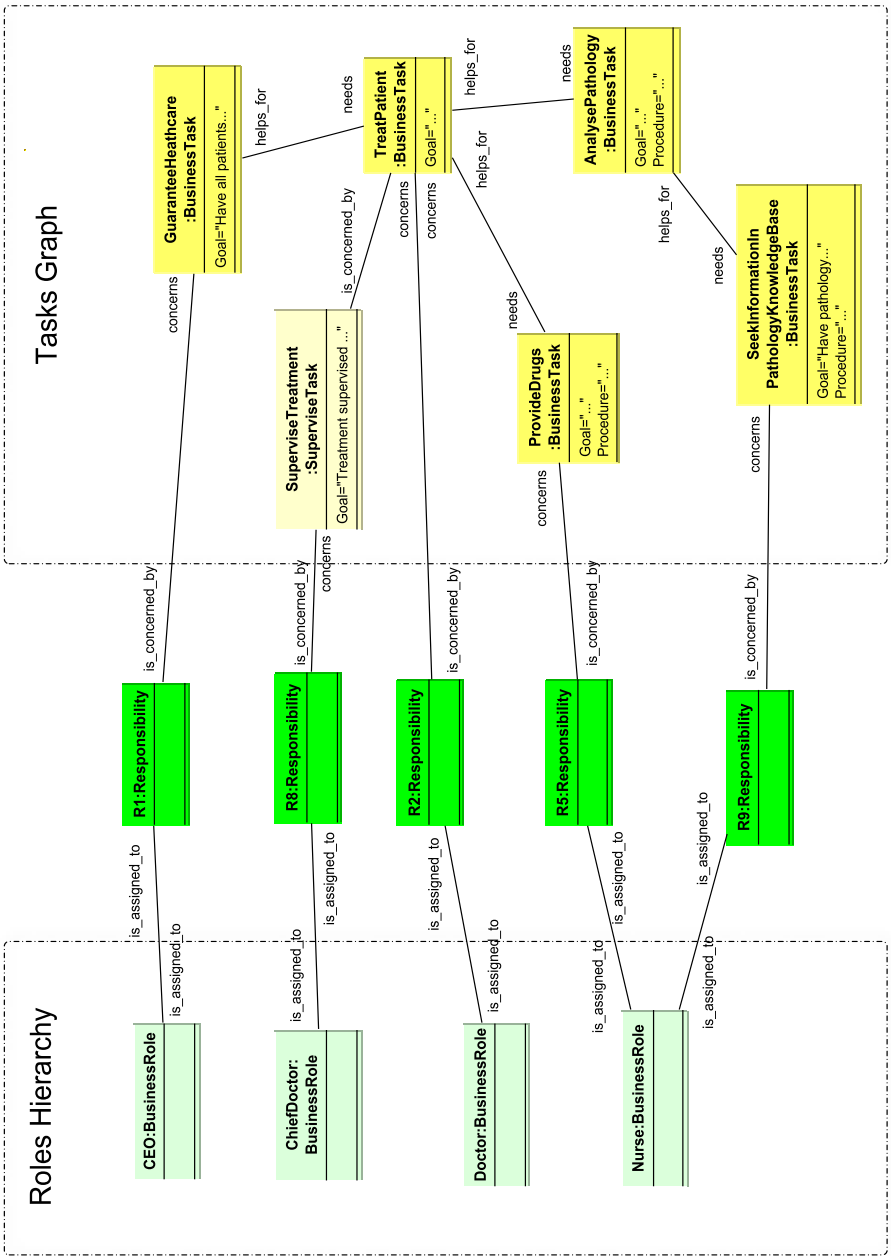


Figure 4.3: Parallel between roles hierarchy and tasks graph

For Orenstein (1999), a business object is an object representing some concepts relevant to the application, whose state is obtained from an underlying relational database. For Caetano et al. (2005), *business objects are [...] representations of organisational concepts, such as resources and actors, which collaborate with each in order to achieve business goals.* These objects exhibit different behaviours according to each specific collaboration context. This means that the perception of a business object depends on its collaboration with other objects. In the Responsibility metamodel, we will not consider (such as advocated by Caetano et al.) that an actor is a business object. This actor is a concept of the metamodel by itself (Figure 4.1).

In ArchiMate, a business task is connected to a business object (Lankhorst (2004)) and both informational concepts are relevant to determine a business domain. The latter is defined in Togaf 9 (The Open Group (2009)) as *a grouping of coherent business functions and activities (in the context of a business sector) over which meaningful responsibilities can be taken.* For Engels et al. (2008), *business objects are the most relevant top–level items of an enterprise.* A business object is a type of an intelligible entity being an actor inside the business layer. In ArchiMate 2.0 specifications (The Open Group (2012)), a business object is defined as *a passive element that has relevance from a business perspective. Business objects represent the important "informational" or "conceptual" elements in which the business thinks about a domain. Generally, a business object is used to model an object type, of which several instances may exist within the organisation. A wide variety of types of business objects can be defined. Business objects are passive in the sense that they do not trigger or perform processes.*

ArchiMate also explains that *business objects may be accessed by a business process, function, a business interaction, a business event, or a business service and may be realised by a representation or by a data object (or both).*

We define the business object, according to our review, as:

> **DEFINITION 2: The *business object* is a passive element (information or document) which has relevance from a business perspective and which may be used by one or many task(s).**

According to this definition, we consider that a business object may cover a large spectre of objects such as an information, a salary, manpower, a room, a car, and so forth.

In ReMMo, we model this business object as a `BusinessObject` class and, because we have reviewed in the above analysis that a task uses business objects, we define the link between the `BusinessObject` and the `Task` as a "use" association such that one `Task uses` zero to many `BusinessObject` and, inversely, one `BusinessObject is used` by zero to many `Task`. This link means that we consider that a `Task` needs to use a `BusinessObject` during its execution.

4.4.5 Example of Task and Business Object modelling

The example of Figure 4.4 illustrates a graph of `BusinessTasks` and `StructuralTasks` from the healthcare domain.

In this example, the `GuaranteeHealthCare` is `BusinessTask` that corresponds to a `Goal` in the healthcare domain. This `BusinessTask` has only a `Goal` attribute (but no `Procedure` attribute) which is to "have all the patients treated". That means that it may be achieved by different ways. This is also the case of the `BusinessTask TreatPatient` that helps to achieve the `BusinessTask GuaranteeHealthcare` and of the `BusinessTask DoX-RayAnalysis` that helps to achieve the `BusinessTask AnalysePatology`. All other `BusinessTasks` include a `Procedure` attribute that indicates how to achieve the `Goal`. This is the case of the `HireEmployee`, `ProvideDrugs`, `AnalysePathology` or `SeekInformationInPathologyKnoweledgeBase`. One isolated `BusinessTask`, `UpgradePatologyKnowledgeBase`, is not attached to the graph. This `BusinessTask` has, i.e., a `Goal` attribute which is "knowledge base always up to date" and a `Procedure` attribute which is "verify and install last version".

The `BusinessTask SeekInformationInPathologyKnowledgeBase` as illustrated in Figure 4.4 uses the `PathologyKnowledgeBase BusinessObject`. This means that the employee who seeks information about the pathology in the knowledge base has to use this pathology knowledge base to do research on the existing pathology and to retrieve the pathology that corresponds to the patient symptoms.

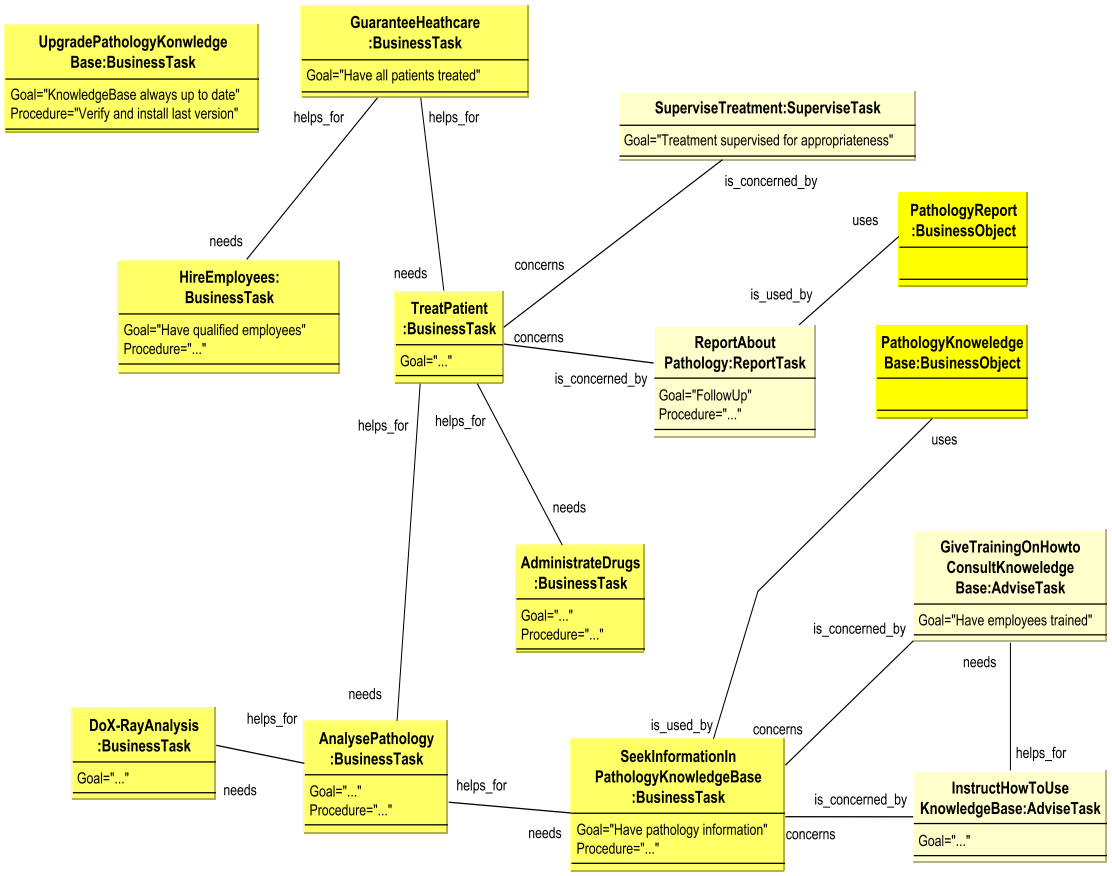


Figure 4.4: Task instantiation in healthcare domain

`TreatPatient` is concerned by the `StructuralTasks ReportAboutPathology` which is of type "Report" and to `SuperviseTreatment` which is of type "Supervise". `ReportAboutPathology` is a task which is defined by a goal to have a `PathologyReport` and with a procedure that expresses the different steps of the reporting procedure. `SuperviseTreatment` is a task with a goal to supervise the treatment but the way to supervise is not defined.

`SeekInformationInPathologyKnowledgeBase` is concerned by the `StructuralTask GiveTrainingOnHowToConsultKnowledgeBase` which is of type "Advise" and that needs, to be achieved, the achievement of the `StructuralTask InstructhowToUseKnowledgeBase`. The latter also concerns the `ConsultPatologyKnowledgeBase BusinessTask`.

4.5 Responsibility and Accountability, Actor, Sanction and Condition modelling

In this section, we introduce the concepts of responsibility and accountability, actor, sanction and condition. These concepts are represented in Figure 4.5.

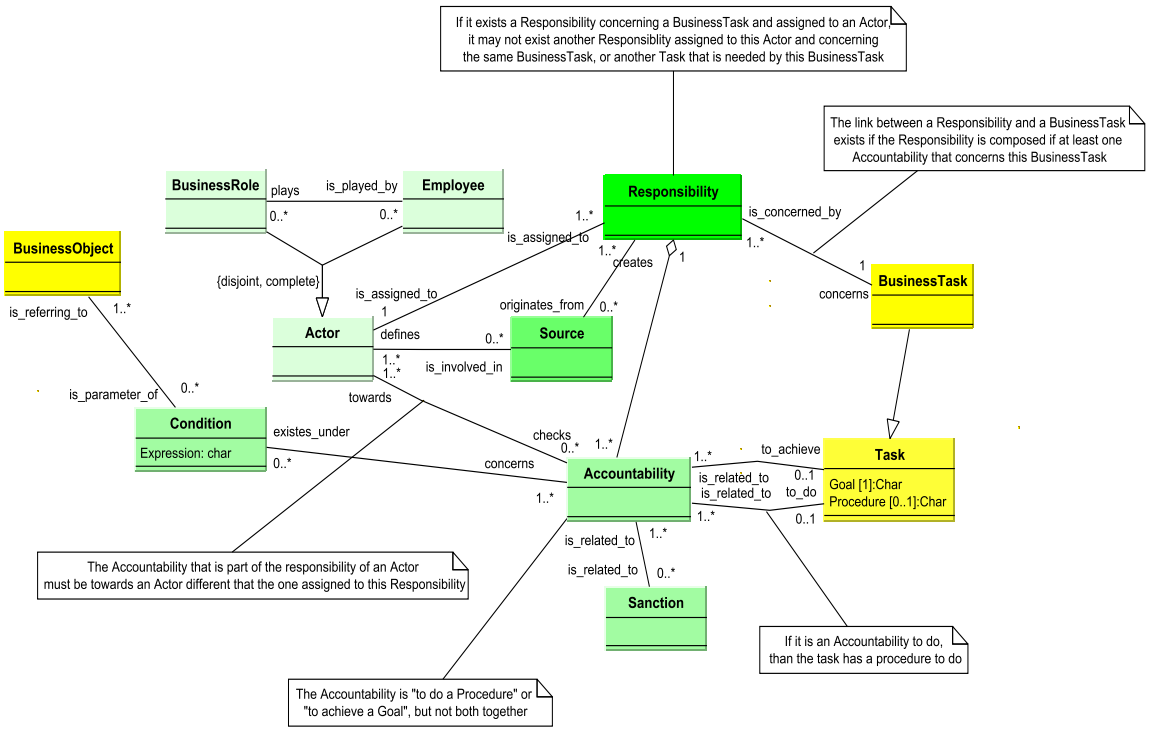


Figure 4.5: Responsibility, accountability and actor modelling

4.5.1 Responsibility and Accountability

As mentioned in Section 3.5.3, Martin et al. (2005) is one of the first authors to have addressed the responsibility in IT by highlighting the problem that arises when the responsibilities of the stakeholders are not suitably addressed. Globally, most of the authors acknowledge that defining the responsibility aims at conferring one or more obligation(s) to an actor (the responsibility owner) (Strens and Dobson (1993), Sommerville (2007a), Storer and Lock (2008)). As a consequence, that obligation provokes a moral or formal duty, in the mind of the responsibility owner, to justify the performance of the obligation to someone else (Strens and Dobson (1993), Prendergast (1995), Cholvy et al. (1997), Stahl (2006), Sliwka (2006), Sommerville (2007a)). Vincent (2011) proposes a structured taxonomy of the responsibility (STRC) and introduces six different perceptions of it according to different science disciplines (Section 3.5.1) and points out the *role responsibility*. The latter means that to improve the management of the responsibility assignment to employee, a set of responsibilities may be assigned to a business role. This statement is also supported by Sommerville (2007a).

Table 4.2 summarises a set of existing definitions of the responsibility through the literature in different domains.

Authors	References	Fields	Definitions
Sommerville, Storer, Lock	Sommerville et al. (2009b)	*Computer science*	A duty, held by some agents, to achieve, maintain or avoid some given state, subject to conformance with organisational, social and cultural norms
Storer, Lock	Storer and Lock (2008)	*Computer science*	Responsibilities are the duties to be discharged by agents
Stahl	Stahl (2006)	*Computer science*	Responsibility is the ascription of an object to a subject rendering the subject answerable for the object
Sliwka	Sliwka (2006)	*Economic*	A superior holds a certain subordinate responsible for a task, when he/she announces his/her beliefs that this subordinate contributes most to this task
Prendergast	Prendergast (1995)	*Law, Economics and Organisation*	The responsibility of an agent is defined as the subset of tasks allocated to him by a manager and it is shown that rent seeking considerations lead the manager to allocate the few tasks to the agent

Table 4.2: Responsibility literature review

Beside the literature related to the responsibility, the review of the literature related to the accountability (Section 3.5.2) highlights the following: Spinello (1996), Mulgan (2000), Laudon and Laudon (2001) and Stahl (2006) express that the responsibility is associated to accountabilities regarding a business task. An accountability is broadly defined as: *the obligation to give account to someone else* (Day and Klein (1987), Sinclair (1995), Erkkilä (2007), Bovens (2010) and Blind (2011)) under the threat of sanction(s). Ackerman (2005) explains that the accountability is a process of justifying [...], the behaviour and the results and to sanction accord-

ingly. Bovens (2007) and Mulgan (2000) explained that the sanction may be positive or negative.

As the concept of accountably appears to be narrowly associated to the concept of obligation regarding a task (Day and Klein (1987), Sinclair (1995), Erkkilä (2007), Bovens (2010) and Blind (2011)), we have also analysed the meaning of this obligation and we acknowledge that it represents what must be done to achieve an expected result (Dobson and Martin (2006)).

Table 4.3 provides a summary of definitions of the accountability reviewed in section 3.5.2.

Authors	References	Fields	Definitions
Ackerman	Ackerman (2005)	*Social sciences*	Accountability is a pro–active process by which public officials inform about and justify their plans of action, their behaviour and results and are sanctioned accordingly
Laudon, Laudon	Laudon and Laudon (2001)	*Management*	Accountability is a feature of systems and social institutions – mechanisms to determine who took responsibility
Goodpaster, Matthews	Goodpaster and Matthews (1982)	*Management*	Accountability is a mechanism set allowing such tracing of causes, actions, and events
Fitzpatrick	Fitzpatrick (2006)	*Management*	Accountability refers to the obligation to demonstrate and take responsibility for performance in light of agreed expectations, and answers the question: Who is responsible to whom and for what?
CobiT	IT Governance Institute (2007)	*Computer science / management*	The accountable is the employee that provides direction and authorises actions
Sommerville, Lock, Storer, Dobson	Sommerville et al. (2009a)	*Computer science*	Obligation to report the achievement, maintenance or avoidance of some given state to an authority
Stahl	Stahl (2006)	*Computer science*	Accountability describes the structures, which have to be in place to facilitate responsibility
Cholvy	Cholvy et al. (1997)	*Computer science*	Accountability is an obligation or a moral duty to report or explain the action or someone else's action to a given authority
Spinello	Spinello (1996)	*Computer sciences*	Accountability is a necessary but not a sufficient condition to be responsible

Table 4.3: Accountability literature review

According to the governance needs of Table 3.1, we acknowledge the need of dealing with the responsibility and the accountable. Therefore, we first model the responsibility concept by a

`Responsibility` class in the Responsibility metamodel. Moreover, the review of the literature and of the governance needs in Chapter 3 allows us to also argue that:

- Responsibility is originated from professional norms and frameworks. This point has been observed in the review of the governance standards and norms in Section 3.3. To trace the origin of the responsibility, we create a `Source` class and we associate it to the `Responsibility` such that one `Responsibility originates_from` zero to many `Source` and one `Source creates` one to many `Responsibility`.
- Responsibility is composed of duties (Storer and Lock (2008) and Sommerville et al. (2009b)) or obligations (Strens and Dobson (1993) and Sommerville (2007a)) and the agent assigned to a responsibility is answerable for the duties (Strens and Dobson (1993), Prendergast (1995), Cholvy et al. (1997), Stahl (2006), Sliwka (2006), Sommerville (2007a) and 3rd sense from Vincent (2011)). To consider these two points, we formulate (1) that a responsibility `aggregates` accountabilities and we model this by linking the class `Responsibility` and `Accountability` and (2) that this accountability corresponds to an obligation which must be justified to someone else.
- Responsibility concerns an object (Stahl (2006)), a task Prendergast (1995) and (Sliwka (2006)) or the achievement, maintenance, or avoiding of some given state (Sommerville et al. (2009b)). Some authors consider that the responsibility concerns a unique business task (Kreifelts et al. (1993), Wang (1999) and Sliwka (2006)) and others, a set of them (Prendergast (1995) and Wang (1999)). In practice, the responsibility for a business task is a state that is delegable from one actor to another, and against which an actor may be evaluated (see actor definition is Section 4.5.2).

Based on these considerations, we define the responsibility as follows:

> **DEFINITION 3: The *responsibility* is a charge assigned to a unique actor to signify its accountabilities concerning a unique business task.**

In ReMMo, we consider that one responsibility concerns a unique business task rather than a set of them. On the contrary, many responsibilities may concern the same business task. Moreover, as a structural task always concerns a business task, we also consider that if a responsibility aggregates an accountability which is related to a structural task, this responsibility concerns the business task which is concerned by this structural task. This means, for example, that if the `Nurse` is `Accountable to_do` the `BusinessTask TreatPatient` and if the `Doctor` is `Accountable to_do` the `ApproveTask` which concerns the `BusinessTask TreatPatient`, both are `Responsible` of the `BusinessTask TreatPatient`.

Based on this, we create a link between the `Responsibility` class and the `BusinessTask` class such that one `Responsibility concerns` one `BusinessTask`, and inversely, such that one `BusinessTask is concerned by` one to many `Responsibility`.

The Responsibility metamodel includes *constraints*, written in natural language, which aims at providing modelling restrictions. Theses *constraints* appear as *notes* in the schemas

of this chapter, and in pink frame in the text.

Accordingly, we formulate that the existence of the link between the `BusinessTask` and the `Responsibility` is function of the following constraint:

> **Constraint I**: *The link between a* `Responsibility` *and a* `Business Task` *exists if the* `Responsibility` *aggregates at least one* `Accountability` *that concerns this* `Business Task`

- Responsibility is assigned to a stakeholder (Martin et al. (2005)), an agent (Prendergast (1995), Storer and Lock (2008) and Sommerville et al. (2009b)), a subject (Stahl (2006)) or to a role (Sommerville (2007a) and 2nd sense from Vincent (2011)). This assignment of responsibility is addressed in Section 4.5.2.
- Responsibility aggregates all the accountabilities of one actor regarding a task. As the task may need sub–tasks, we consider that the responsibility of an actor also includes the accountability of that actor related to the sub–tasks of that task. Therefore we introduce the constraint that:

> **Constraint II**: *If it exists a* `Responsibility` *concerning a* `Business Task` *and assigned to an* `Actor`*, it may not exist another* `Responsibility` *assigned to this* `Actor` *and concerning this* `Business Task`*, or another* `Task` *that is needed by this* `Business Task`

Secondly, we define the accountability as follows:

> **DEFINITION 4: The *accountability* is an element which is part of a unique responsibility and/or which represents an obligation of an actor to achieve the goal, or to perform the procedure of a task, and the justification that it is done to someone else, under threat of sanction.**

And in like manner, we introduce an `Accountability` class in the Responsibility metamodel and argue that:

- `Accountability` is related to a `Task` (Spinello (1996), Mulgan (2000), Laudon and Laudon (2001), Stahl (2006), Bovens (2007) and 3rd sense from Vincent (2011)). More precisely, an accountability is characterised by a result or behaviour that should be achieved (a task) (Ackerman (2005)). Therefore, we have created two links between the `Accountability` and the `Task`. The first link signifies that an `Accountability` is to_do zero or one `Procedure` of the `Task` (2nd definition from Cholvy et al. (1997), equivalent to the causal responsibility of Doing from Sommerville (2007b)) and inversely that the `Procedure` of a `Task` is_related_to one or many `Accountability`. The second link signifies that one `Accountability` is to_achieve zero to one `Goal` of the `Task` (3rd definition from Blyth et al. (1993) and Cholvy et al. (1997), and, inversely, that the `Goal` of a `Task` is_related_to one or many `Accountability`. As the to_do link only exists if a `Procedure` is defined, we introduce the following constraint that:

Constraint III: *If an* `Accountability` *is related to a* `Task` *with a to_do link, then the attribute* `Procedure` *is defined for the* `Task`

Moreover, to guarantee that the accountable always refers either to the achievement of a goal or to the performance of a task, we introduce the constraints that:

Constraint IV: *The* `Accountability` *is either to_do a* `Procedure` *or to_achieve a* `Goal` *but not both*

- Accountability aims to make account to someone else (Day and Klein (1987), Sinclair (1995), Erkkilä (2007), Bovens (2007), Bovens (2010) and Blind (2011)). To consider this point in the metamodel, we create a link between the `Accountability` and the `Actor` such that one `Accountability` is towards one to many `Actor` (Erkkilä (2007)) and, inversely, that one `Actor` checks zero to many `Accountability`. As the check of the accountability has as objective to analyse whether an accountability has been fulfilled or not, and to sanction or not accordingly, the accountability must be towards one actor different than the actor assigned to the responsibility which aggregates the accountability. We therefore introduce the constraint that:

Constraint V: *The* `Accountability` *that is part of the* `Responsibility` *of an* `Actor` *must be towards one* `Actor` *different than the one assigned to this* `Responsibility`

- In practice, we have observed that some accountabilities composing a responsibility may not apply under certain conditions. Therefore, we create a link between the `Accountability` class and the `Condition` class which is explained in Section 4.5.6.
- Accountability may result in sanctions (Mulgan (2000), Ackerman (2005), Bivins (2006), Sommerville (2007a), Bovens (2007), Dubnick (2007), Fox (2007) and Bovens (2010)). To consider this point, a class `Sanction` that is aggregated to the class `Accountability` has been integrated. Section 4.5.5 reviews the semantic of the sanction.

4.5.2 Actor

The responsibility is defined for a unique actor to which it is assigned. The concept of actor has already been largely defined in the literature and it will not be reviewed in detail in this work. This concept of actor has been defined in i* as an active entity which carries out actions to achieve goals by exercising its know–how. i* uses the term actor to refer generically to any unit to which intentional dependencies can be ascribed and argues that it may be either a role, an agent or a position (the latter being an intermediary between the role and the agent (Amyot et al. (2009)).

Such as in the i* specification, we consider that the actor is a generic entity that may be of different types and we define this actor as:

DEFINITION 5: The *Actor* is an active entity which is assigned a set of responsibilities and that may check accountabilities.

In ReMMo, we create an `Actor` class and we associate it to the `Responsibility` such that: one `Responsibility is assigned to` one `Actor`. Inversely, the `Actor` may be assigned to one or many `Responsibility`. Therefore, we define the link one `Actor is assigned to` one to many `Responsibility`.

We also keep the role and the agent as type of actors but we rename the term agent in employee and the role in business role, thereby rendering the metamodel more business oriented. Additionally, although it is possible to assign responsibility either for an employee or for a business role, these responsibilities must be assigned to business roles as much as possible to reduce the number of actors to be defined. However, in some cases where it is necessary to highlight a specific responsibility of a unique employee, it may be required to defined and assigned a responsibility directly for this employee. This is the case for instance when the employee has specific responsibilities that may not be shared with a role.

4.5.3 Business Role

The notion of role has been widely addressed by Fuchs et al. (2011) who have conducted a wide state of the art related to this element. According to RBAC (Ferraiolo et al. (2001)), a role is *a job function within the context of an organisation with some associated semantics regarding the authority and responsibility conferred on the user assigned to it.* According to ArchiMate 2.0 specifications (The Open Group (2012)), a business role is defined as *the responsibility for specific behaviour which can be assigned to an actor.* Moreover, the framework argues that the business processes or business functions are assigned to a unique business role with certain responsibilities or skills. In addition to the relation of a business role with a behaviour, a business role is also useful in a (structural) organisational sense; for instance, in the division of labour within an organisation. i* explains that the role is an abstract characterisation of the behaviour of a social actor within some specialised context or domain of endeavour. Its characteristics are easily transferable to other social actors. The dependencies associated with a role apply regardless of the agent who plays it.

We define the business role as:

DEFINITION 6: The *BusinessRole* is a type of actor which represents a set of employees who share common characteristics.

In the Responsibility metamodel, we introduce a `BusinessRole` class that is a type of `Actor` and we consider that a `BusinessRole` is specified in the context of an organisation and is assigned to employees having the same position in the company, due to their hierarchy (e.g., a unit manager, a secretary,...), due to their education (e.g., doctor, nurse,...), due to their experience (e.g., senior engineer, expert in a precise pathology,...), or/and due to their domain of competence (e.g., IT department, pulmonary, accounting department,...). We use the term business role since this role is defined according to business attributes, is related to business tasks and to employees.

4.5.4 Employee

In IT, an employee is defined as the entity that provides a service or that realises an activity, a business task or a process and it may have different names depending on the concerned field. In RBAC for instance, that entity is called *User* and is defined as a human being although the authors acknowledge that this concept of user can be extended to include machines, networks, or intelligent autonomous agents. In CIMOSA, this entity is called an *Agent* and in ArchiMate, it is referred to as the Business Actor and is defined as *an organisational entity that is capable of performing behaviour.* Moreover, a business actor that is assigned to a business role ultimately performs the corresponding behaviour. In i*, the *Agent* is an actor with concrete, physical manifestations, such that a human individual. i* uses the term agent instead of person to be generic enough, so that this concept may be used to refer to human as well as to a hardware/software agents. An agent has dependencies that apply regardless of which roles he/she/it is playing.

We define, based on this analysis, the employee as :

DEFINITION 7: The *Employee* is a type of actor which represents a human entity which may or may not play, one or more business roles.

In ReMMo, we introduce an `Employee` class as a subclass of `Actor` and we consider that this class is associated to the class `BusinessRole` such that one `Employee plays` zero to many `BusinessRole` and, inversely, such that a `BusinessRole is_played_by` zero to many `Employee`.

4.5.5 Sanction

Bovens (2007) introduces the sanction in his definition of accountability acknowledging that *an actor may face consequences* resulting to the appreciation of the achievement of its accountabilities. Mulgan (2000) also considers that the sanction is a component of the accountability although Fox (2007) considers the hard accountability when there exists sanction(s) and soft accountability when there is not sanction. In that case, an accountability is equivalent to an answerability as introduced by Stahl (2006). For Dubnick (2007), the sanction *[...] can act as a background reminder for the actor about its moral engagement.* Mulgan (2000) and Bovens (2007) consider that the sanction may be positive or negative: *Positive sanctions are for instance a reward, recognition, the receipt of an amount of money although the negative sanction can be a disciplinary measure, a civil remedy.* Sommerville (2007a) also used the twofold of sanctions for its works.

Authors	References	Fields	Definitions
Ackerman	Ackerman (2005)	*Social sciences*	It is a pro–active process by which public officials inform and justify their plans of action, their behaviour and results are sanctioned accordingly
Bovens	Bovens (2007)	*Legal sciences*	Sanction is a constitutive element of the [...] accountability. An actor is formally or informally subject to sanction in case of bad performance or to rewards in case of adequate performance

Authors	References	Fields	Definitions
Sommerville	Sommerville (2007a)	*Computer science*	A negative sanction is blame received for the occurrence of some state of affairs
Sommerville	Sommerville (2007a)	*Computer science*	A positive sanction is credit or praise received for the occurrence of some state of affairs

Table 4.4: Sanction literature review

Based on this analysis, we provide our own definition of the sanction which is:

DEFINITION 8: The *Sanction* is an element associated with an accountability and which corresponds to the consequence resulting from the justification of the realisation (or not) of this accountability.

We consider that a `Sanction` may be associated to one or more `Accountability`. Therefore, we express that one `Sanction is_related_to` one to many `Accountability` and that, inversely, an `Accountability is_related_to` zero to many `Sanction`. Thereby, we assume that if an `Accountability` is associated with no `Sanction`, it corresponds to an answerability according to Fox (2007).

4.5.6 Condition

We have reviewed in Section 4.5.1 that responsibilities are defined relatively to a unique task and for a unique actor. In practice, we have noted that those responsibilities may evolve with the context. For instance, the `Responsibility` of a `Doctor` to `TreatPatient` may aggregate the `Accountability to_do SeekInformationInPathologyKnowledgBase` if this `Doctor` is in the hospital although this `Accountability` may be part of the `Responsibility` of the `Nurse` if the `Doctor` is absent from the hospital.

To represent this in the Responsibility metamodel, we introduce a `Condition` class that provides a set of rules that governs the existence of the accountabilities and which are dependent on the context. This context is represented in ReMMo by some value of one or more business object(s). For instance, in the above case, the presence or absence of the doctor in the hospital is a business object and the `Accountability to_do SeekInformationInPathologyKnowledgBase` that is part of the `Responsibility` of the `Nurse` is applicable if the `Doctor` is absent, acknowledging that this absence is modelled with the state of business object.

We define the condition as:

DEFINITION 9: The *Condition* defines a context which must be verified for the accountability to exist.

In the Responsibility metamodel, we express that the `Condition` is dependent on the `BusinessObject`. Accordingly, we represent that one `Condition refers_to` one to many `BusinessOb-`

`ject` and inversely, one `BusinessObject is_parameter_for` zero to many `Condition`. We associate the `Condition` to the `Accountability` such that one `Accountability exists_under` zero to many `Condition` and inversely, one `Condition concerns` one to many `Accountability`.

Among the conditions that we may model, we find:

4.5.6.1 The separation of duties.

As explained in Sandhu (1990), the separation of duties aims at limiting the authority given to a unique employee preventing him to perform fraudulent tasks. This is by the way a business constraint that, until now, have been faced by many mechanisms at the application layer (Finin et al. (2008) and Ferrini and Bertino (2009)). Based on the `Condition` class of our Responsibility metamodel, we are able to address this business constraint at the business level by expressing that two accountabilities A and B may not be assigned to the same actor. Therefore, we need to create a condition C that states that if the accountability A is part of a responsibility assigned to this actor, then accountability B may not exist in this responsibility or in any other responsibility assigned to this actor. To manage this, we create two business objects that correspond: (1) to a list of responsibilities and their existing composing accountabilities and (2) to a list of assignments of the responsibility for the actors.

4.5.6.2 The delegation

The delegation corresponds to the transfer of an accountability to_do or to_achieve a task to someone else. E.g., if actor A delegates its accountability to_do a task T to actor B, the accountability to_do the task T no longer exists in the responsibility assigned to actor A but exists in the responsibility assigned to actor B. At the same time, a new accountability may appear in the responsibility assigned to actor A which is to_do the supervision of the task T. To manage this, we create a business object that corresponds to the state of the delegation of the accountability to_do the task T from actor A to actor B and based on this state, one condition expresses the existence of the accountabilities of one actor and another condition expresses the existence of an accountability of another actor.

4.5.6.3 Chinese Wall security policy

The Chinese Wall security policy has been proposed by Brewer and Nash (1989a). The principle of this security policy is to limit the access to a business object when another business objects from the same *conflict class* has already been accessed by the same actor. Hence, this *conflict class* represents the set of business object which may be accessed by an actor as long as another business object from this class has never been accessed by this same actor. The Chinese Wall policy provides, as a result, constraints based on past usage of the business object. With the Responsibility metamodel, we may address this policy using the `Condition` class. To manage this, we firstly create a business object which corresponds to the conflict class and a business object which corresponds to the records of the use of the business object from the conflict class by the accountabilities. Secondly we create a condition of existence of the accountabilities (that need to use business objects from the conflict class) which is *No business object from the conflict*

class has been used by the actor. Hence, if the business object A and the business object B are in the same conflict class, as soon as one accountability which is part of the responsibility of the actor A uses the business object A or the business object B, all other accountabilities that require the right to use a business object from this *conflict class* and which are part of this responsibility or another responsibility assigned to this actor A may no longer exist.

4.5.7 Example of Responsibility, Accountability, Actor and Condition modelling

The first responsibilities that we address, in the example of Figure 4.6, are the ones of the `Doctor` which are concerned by the task `SeekInformationInThePathologyKnowledgeBase`. This `Task` is described by a `Goal` that is related to the `Accountability AC11`. This `AC11` is part of the `Responsibility R2`. The `Task` is also described by a `Procedure` that is related to two `Accountability` which exist under the `Condition C2` *Doctor in Hospital* or the `Condition C1` *Doctor not in Hospital*, and which both refer to the `BusinessObject PresenceOfDoctorInHospital`. The `AC1` is part of `R2` assigned to the `Doctor` and exists if `Condition C2` is true. The `AC19` is part of `R19` assigned to the `Nurse` and exists if `Condition C1` is true. One difference between both accountabilities is that `AC1` is towards the `DoctorGeneral` although `AC19` is towards the `Doctor`. In this second case, the `Doctor` is no longer accountable to do `SeekInformationInThePathologyKnowledgeBase` but he remains accountable to achieve `SeekInformationInThePathologyKnowledgeBase` (`AC1`) towards the `DoctorGeneral`.

To upgrade the `PathologyKnowledgeBase` is a very critical task and, to avoid too many employees to do it, a dedicated and well identified employee has therefore been designated: `Alice`.

Figure 4.7 represents the modelling of a delegation of the task `TreatPatient` from the `Doctor` to his `Assistant`. When a `Doctor` delegates this task, the delegation is recorded in the `BusinessObject DelegationState`. In the function of this `BusinessObject`, if the condition `DelegationCondition1` *The doctor has delegated the treatment of the patient to his assistants* is true, `AC2` and `AC3` exist. Inversely, if the condition `DelegationCondition2` *The doctor has not delegated the treatment of the patient to his assistants* is true, `AC1` exists.

4.6 Capability and Right modelling

Capability and rights are components that already exist in the field of IT and which we have introduced in ReMMo. These concepts are represented on Figure 4.8.

4.6.1 Capability

To realise his accountability, an actor must possess a set of capabilities. These capabilities are intrinsic to the actor and correspond to the knowledge, the know–how, or the attitude he possesses. The capabilities have been analysed by Vernadat (2002). Among the most used capabilities, according to our observation of real situations, we retrieve:

- The education. E.g., the CEO must have a manager education (Figure 4.12)
- The experience and the knowledge about the enterprise. E.g., the Doctor General must have a previous experience in the healthcare domain.

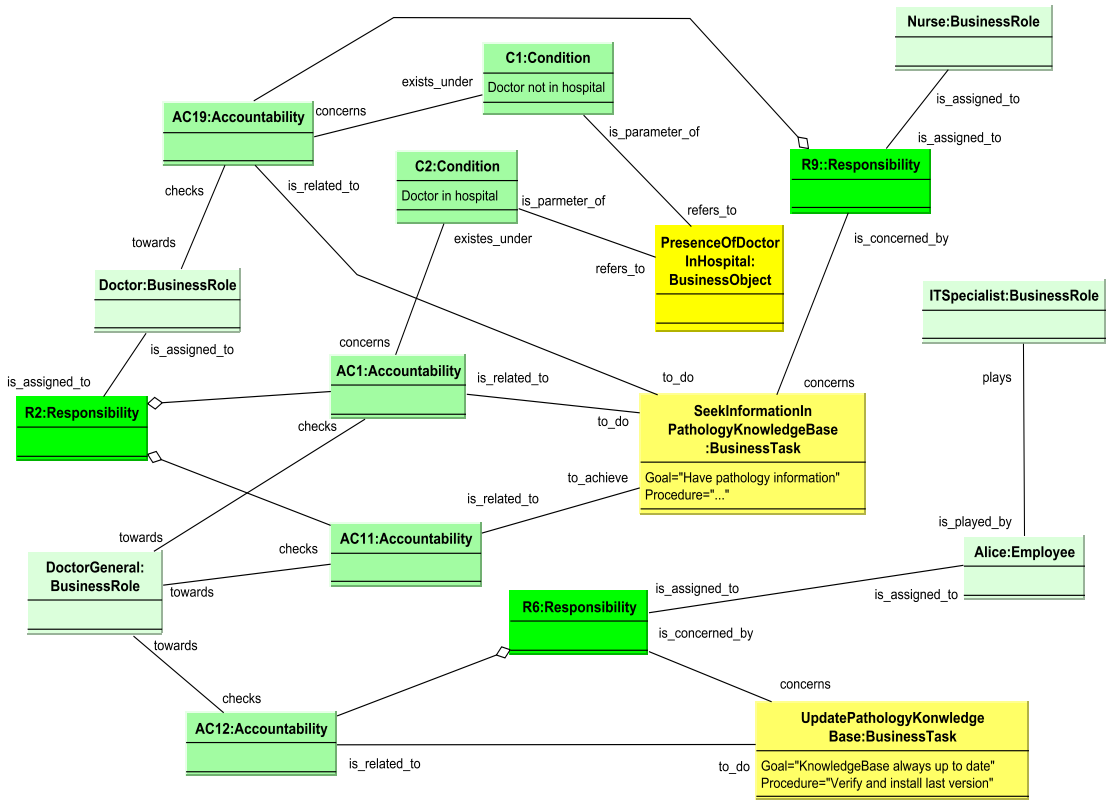


Figure 4.6: Responsibility instantiation in healthcare domain

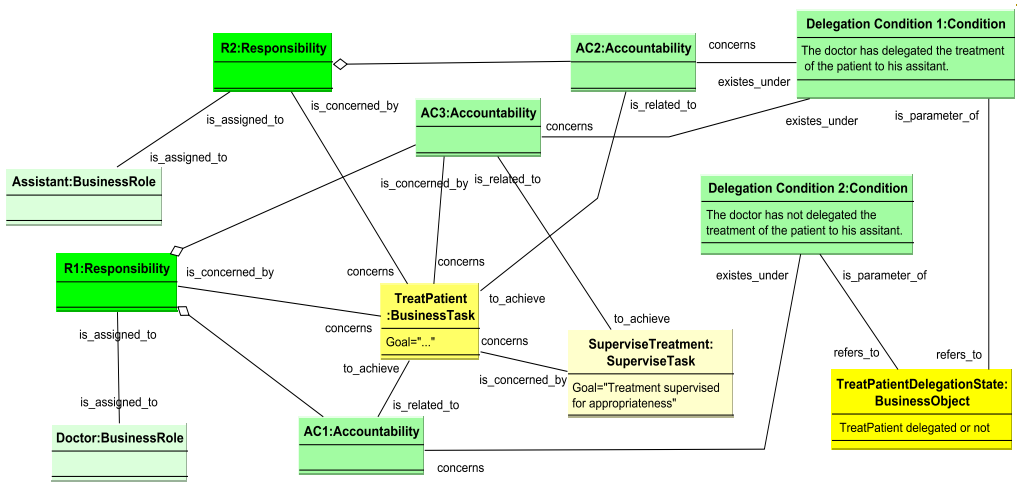


Figure 4.7: Example of delegation in healthcare domain

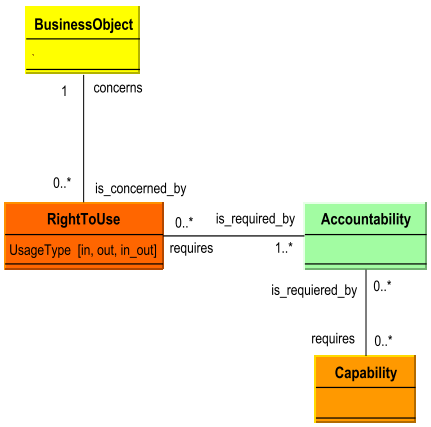


Figure 4.8: Capability and rights to use modelling

- The human being. E.g., the CEO is open–minded and communicative with the doctors.
- The authority. It describes *the power or right to give orders or makes decisions, to command and control other employees and to assign responsibilities* following Zelm et al. (1995) and it is defined in Sommerville (2007a) as *management authority over others to ensure that the tasks necessary to discharge the responsibility are completed.* The CIMOSA meaning of authority includes two aspects: firstly, the authority intrinsic to the employee or the authority that he receives because of his assignment to responsibility. That last type is considered as a right rather than a capability since that authority is given by the enterprise rather than the ability possessed inherently by the doctor.
- The ability to perform a business task. Globally, the ability to perform a business task is obtained after training or due to the education. For instance, *Alice* has a nursing education (and is thus capable of performing nursing tasks) and *doctor Bob* a surgical one (and is thus capable of performing surgery tasks).
- The ability to use software. That ability is obtained, e.g., after a dedicated training or due to a previous experience. This ability is hence a specialisation of the ability to perform a task.
- The physical characteristics. E.g., the paediatric nurse must be female.

Based on the upper analyse, we define the capability as:

DEFINITION 10: The *Capability* represents the qualities, the skills or the resources intrinsic to the actor and which are required to perform one or several accountability(ies).

To represent this capability in the Responsibility metamodel, we create a `Capability` class that is associated to the `Accountability` such that one `Accountability` requires zero to many `Capability` and, inversely, such that one `Capability` is_required_by zero to many `Accountability`.

4.6.2 Right To Use

The concept of right is common but is not systematically embedded in all IT frameworks, Vernadat (2002) and IT Governance Institute (2007). It encompasses facilities required by an employee to fulfil his accountability(ies). These facilities are described in terms of access to a business object and may, for instance, represent:

- a right to access information. This access may be to read information (such as consult the patient's files), to create information (such as create a new pathology analysis report) or to modify information (such as update the evolution of a patient's pathology).
- a right to access another business object such as the right to have a salary, to have a company car, to have access to a meeting room, to have the support of a team, and so forth.

Based on this analysis, we define the right to use as:

DEFINITION 11: The *RightToUse* represents an authorisation to perform an operation on a business object which is required to perform one or several accountability(ies).

The `RightToUse` class has been created in the Responsibility metamodel to represent the right. This class is associated to the `Accountability` such that one `Accountability` requires zero to many `RightToUse`, and that, inversely, one `RightToUse` is_required_by one to many `Accountability`. This `RightToUse` class is also associated to the `BusinessObject` such that one `RightToUse` concerns one `BusinessObject` and that, inversely, one `BusinessObject` is_concerned_by zero to many `RightToUse`. Accordingly, we introduce the following constraint:

Constraint VI: *The link between the `Business Object` and the `Task` exists if it exists at least one `Right to Use` the `Business Object` required by the `Task`*

The `RightToUse` is characterised by a `UsageType` which can be of three types:

- The `In` means that the right is, for the accountability, to access the `BusinessObject` as an input.
- The `Out` means that the right required by the accountability is to update or create an output that concerns a `BusinessObject`
- The `In_Out` means that the accountability required the right to access, create and/or to modify the `BusinessObject`

4.6.3 Example of Right and Capability modelling

The `RightToUse` that we illustrate on Figure 4.9 is the right to use the `PathologyKnowledgeBase` to seek information about the patient pathology or to update it. We therefore define two types of rights to use the pathology knowledge base. This first is the right of the type `In` for the `Accountability AC1` and `AC19`. This means that the `PathologyKnowledgeBase` is

used as an input for the `Accountability`. The second is the right of the type `Out` for the `Accountability AC12` that means that the accountability requires the right to update the `PathologyKnowledgeBase`.

On Figure 4.12, we illustrate also that `Accountability AC2` to achieve the `Goal` of `GuaranteeingHealthcare` requires the `Capability` which is to have a `ManagerEducation`.

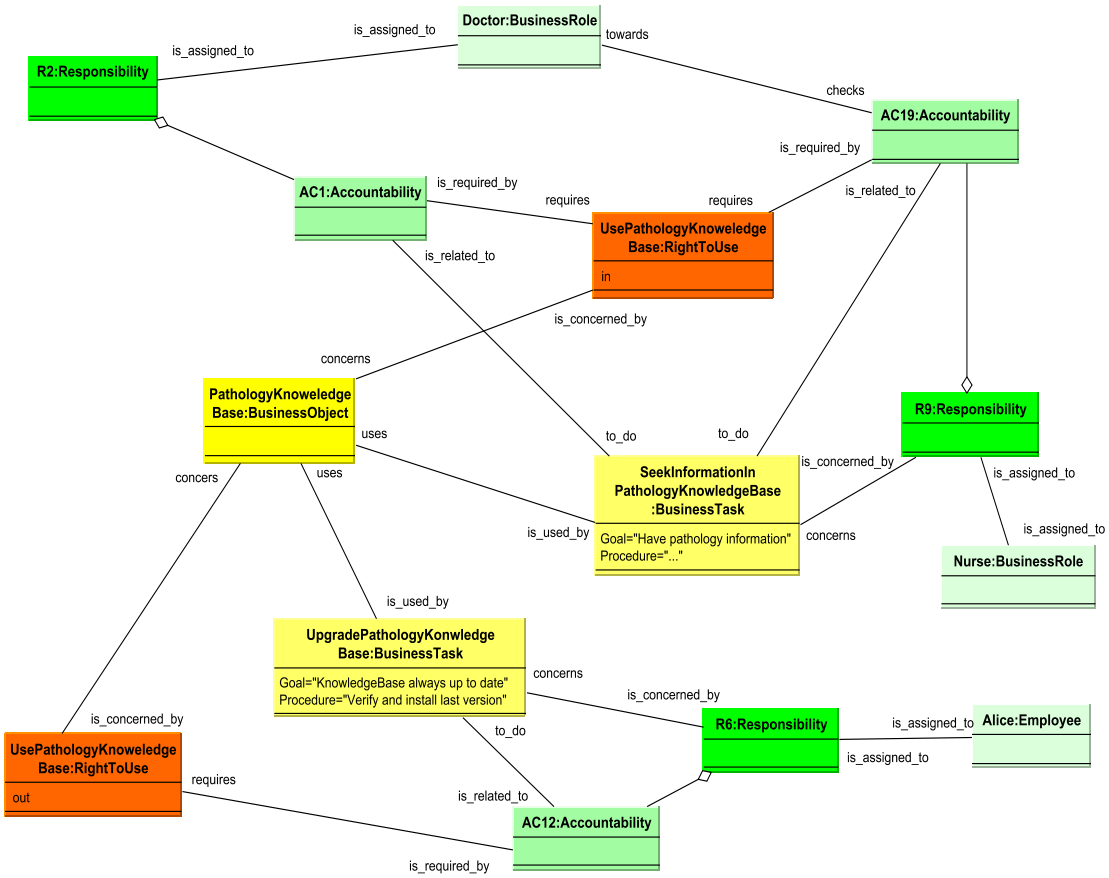


Figure 4.9: Rights instantiation in healthcare domain

4.7 Governance Rules and Source modelling

In this section, we introduce the concepts of governance rule and of source that we represent on Figure 4.10.

In practice, we have observed that these governance rules originated from governance source like the ones reviewed in Chapter 3. These governance sources provide high level rules that impact the elaboration of the responsibilities by expressing conditions over the accountabilities. These rules are, for instance, the Separation of Duties, the delegation rules (e.g., does the delegator keep the accountability to_achieve a goal when he delegates the accountability to_do the

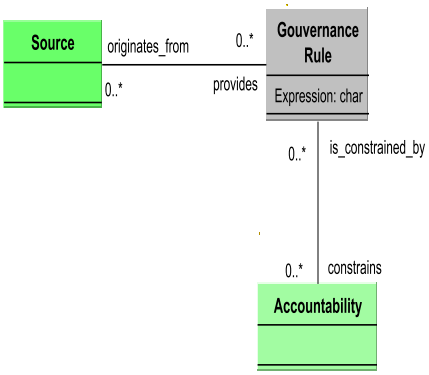


Figure 4.10: Governance rule modelling

procedure of a task?), the Chinese Wall policy, the Two–man rule[1], or principle of mutually–exclusive roles. Another type of governance rule is the one which creates one responsibility, assigned to the management, to manage the responsibilities to be assigned to the other employees.

We have introduced a `GovernanceRule` class in ReMMo to represent the governance rule. This class is associated with the `Source` class such that one `GovernanceRule` originates_from zero to many `Source` and, inversely, such that one `Source` provides zero to many `GovernanceRule`.

The `GovernanceRule` may have an impact on most of the elements of the Responsibility metamodel. However, its final impact is on the `Accountability`. Therefore, for clarity reasons, we only represent the link between the `GovernanceRule` and the `Accountability` such that one `GovernanceRule` constrains zero to many `Accountability` and, inversely, such that one `Accountability` is_constrained_by zero to many `GovernanceRule`.

Based on this analysis, we define the governance rule as:

DEFINITION 12: The *GovernanceRule* is a high level prescript originating from dedicated sources and which constraints the definition of the accountabilities.

And we define the source as:

DEFINITION 13: The *Source* is a formal piece of information which creates responsibilities and which contains, amongst other, required or desired governance rules.

[1]The two–man rule is a control mechanism designed to achieve a high level of security for especially critical material or operations. Under this rule all access and actions require the presence of two authorised people at all times.

4.7.1 Example of Governance Rule modelling

The governance rule that we illustrate in Figure 4.11 is dictated by the `MedicalLaw Source` and imposes that the `Actor` who `ReportsAboutPathology` must be the same as the `Actor` that performs `TreatPatient`. This `GovernanceRule` constrains `AC4` and `AC5` to be part of the same `Responsibility R2`, which is assigned, in our case to the `Doctor`.

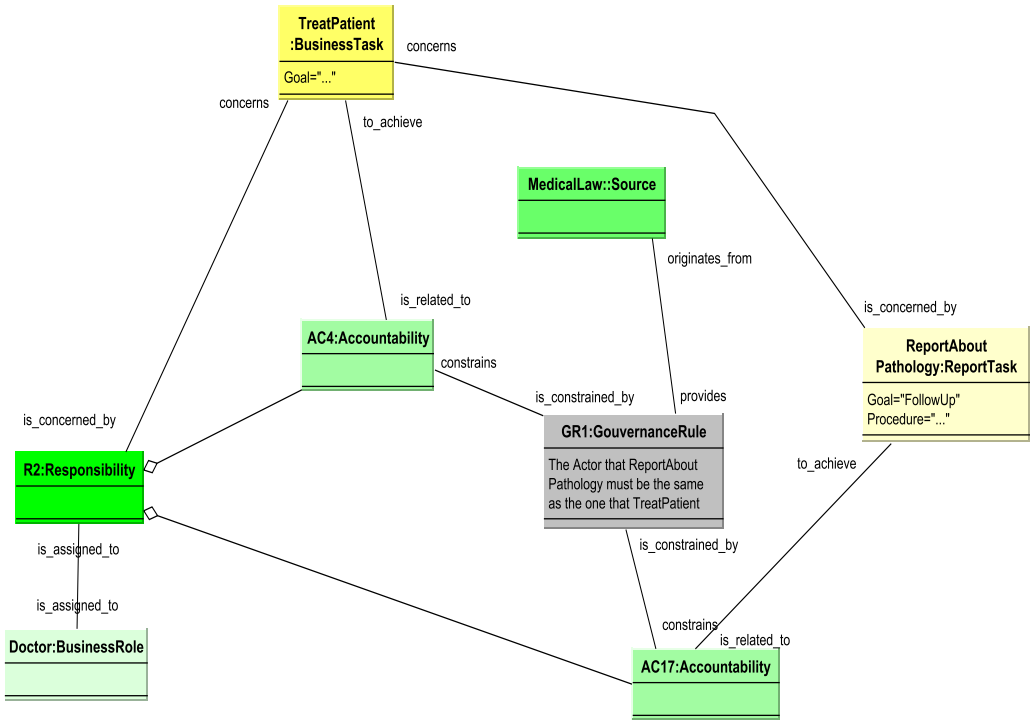


Figure 4.11: Governance rule instantiation in healthcare domain

Figure 4.12 represents the Responsibility metamodel instantiated in the healthcare domain following the case study introduced in Section 4.2.

4.8 Conclusions

In this chapter, we have elaborated a Responsibility metamodel (Figure 4.13) that includes the concepts necessary for the description of the responsibility at the business layer. Each responsibility defined from the metamodel is assigned to an actor and concerns a business.

The accountability is an element that is part of the responsibility and that may concern either the achievement of the goal of a task or the performance of the procedure that allows achieving this goal. This accountability is checked by an actor and sanctions may result from this check. To realise the accountability, rights and capabilities are necessary.

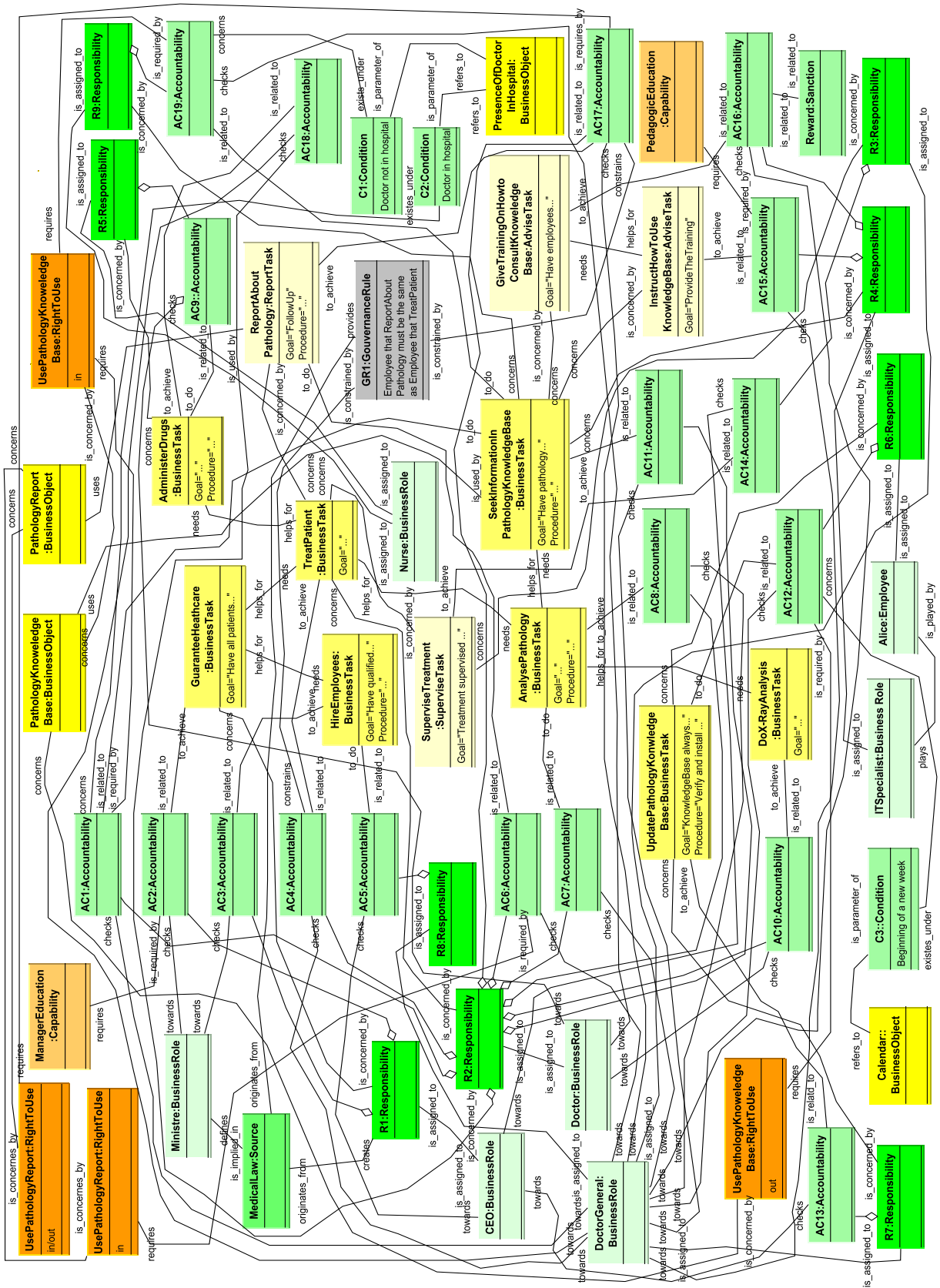


Figure 4.12: Responsibility metamodel instantiated in the healthcare domain

For each of these concepts, we have analysed how it has previously been addressed in the literature related to the information technologies and to other disciplines. This analysis has allowed to integrate the concepts in the Responsibility metamodel and to associate them with the other concepts. We have provided our own definition for each of the concepts and we have illustrated how they may be instantiated based on a case study from the healthcare domain.

Additionally, in Appendix E, we have highlighted how it is possible to model the RACI chart from COBIT, and from alternative RACI models, using ReMMo. We have analysed, e.g., that (R)esponsibility and (A)ccountability correspond respectively to a responsibility which aggregates an accountability to do and to achieve a task, that (C)onsulted corresponds to a responsibility which aggregates an accountability to do or to achieve a structural task of a type advice task, and that (I)nformed corresponds to a right necessary for another accountabilities.

Publications related to this chapter:

- C. Feltus, M. Petit, Building a Responsibility Model Including Accountability, Capability and Commitment, in *Proceedings of the 4th International Conference on Availability, Reliability and Security (ARES)*, Fukuoka, Japan. 2009. IEEE.
- C. Feltus, M. Petit, F. Vernadat, Enhancement of CIMOSA with Responsibility Concept to Conform to Principles of Corporate Governance of IT, in *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, Moscow, Russia. 2009.
- C. Feltus, M. Petit, E. Dubois, Strengthening Employee's Responsibility to Enhance Governance of IT – CobiT RACI Chart Case Study, in *Proceedings of the 1st Workshop on Information Security Governance (WISG CCS)*, Chicago, Illinois, USA. 2009. ACM.
- C. Feltus, E. Dubois, M. Petit, Conceptualizing a Responsibility based Approach for Elaborating and Verifying RBAC Policies Conforming with CobiT Framework Requirements, in *Proceedings of the 3th International Workshop on Requirements Engineering and Law (RELAW10)*, Sydney, Australia. 2010. IEEE.

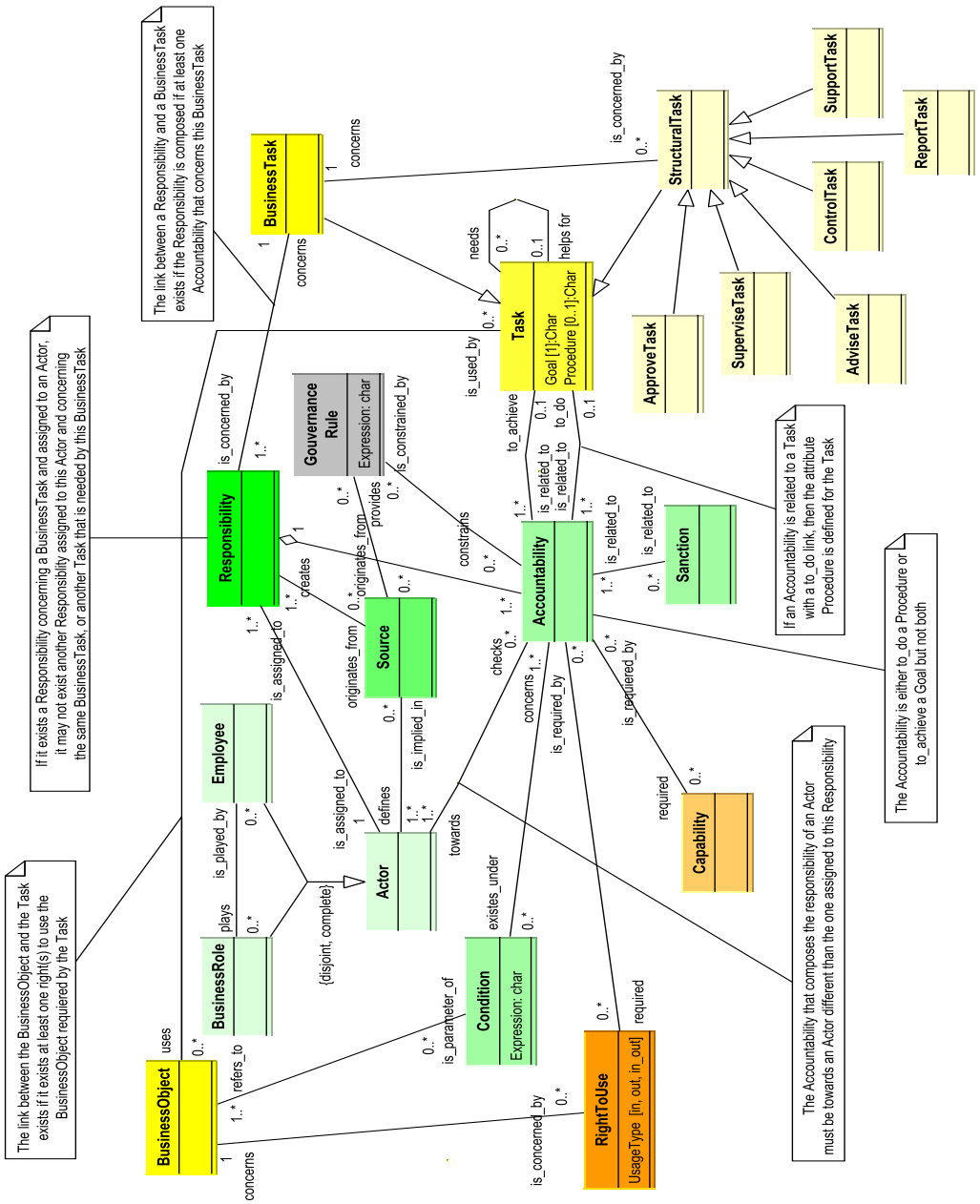


Figure 4.13: Responsibility metamodel

Chapter 5

Conceptual mapping and integration between the Responsibility metamodel and ArchiMate business layer

5.1 Introduction

The enterprise architecture models enable the description, the explanation and the justification of the different elements which compose an enterprise, as well as the connections between these elements. They provide views that are understandable by all the stakeholders and which permit to make business IT alignments knowing the interaction between each enterprise artefacts. For instance, they permit to understand and calculate the impact, from a technical point of view, of a new business service and, as a result, permit to analyse the needed server capacity. In the other direction, the failure of a server has an impact on an application and thus, on the business services.

To support this alignment between the enterprises elements, enterprises architecture management (EAM) had undergone major improvements during the first decade of 2000. Even if the advantages of the enterprise architecture models are no longer to be demonstrated, the high abstraction level of the modelled concepts, and of the associations between these concepts, sometimes make it difficult to accurately use the architecture models to perform, verify or justify the engineering of the access rights, on an application, to be provided to an employee based on his/her responsibilities.

In this chapter, we realise a conceptual mapping between ArchiMate and the Responsibility metamodel in order to integrate both. The resulting integrated metamodel aims at providing an overview on how ReMMo contributes to motivate some elements of the enterprise architecture metamodel and how it enhances the alignment among the ArchiMate elements to support the definition of the access rights provided to the employees according to their responsibilities. This work considers the mapping between the concepts and relations between concepts from both metamodels. Practically, this conceptual mapping integrates ReMMo in ArchiMate while re-

maining compliant with the latter's specifications. Considering this, we are forced not to modify the ArchiMate metamodel nor to enrich it with new concepts, but rather to extend it according to its internal extension mechanism.

This chapter introduces ArchiMate in Section 5.2 and explains the resolution of heterogeneity in Section 5.3. Afterwards, it presents the mapping and integration of the Responsibility metamodel and ArchiMate in Section 5.4 and evaluates it with a case study at the Centre Hospitalier de Luxembourg in Section 5.5.

5.2 Introduction to ArchiMate

5.2.1 ArchiMate overview

ArchiMate is an enterprise architecture metamodel used by the IT architects to design business and IT static views and their links, of the corporate architecture (Lankhorst (2004)). ArchiMate allows reducing the complexity and proposes means to model and thus better understand the enterprise, and the interconnections and interdependency between the processes, the people, the information, and the systems. Consequently, one objective of ArchiMate is to provide pictures of each enterprise architecture aspects such as the organisational structure, the business processes, the information processing system or the infrastructure. It permits to ensure uniform semantics of the instantiated models but it is not really appropriate to enable quantitative analysis.

Another objective of the enterprise architecture is to highlight the creation of business value. For instance, in the Archissurance scenario (Lankhorst (2004)), the customer needs to "be insured" with the instance "be insured" being a type of "business value". This business value is generated by the business processes which are supported by applications and infrastructures.

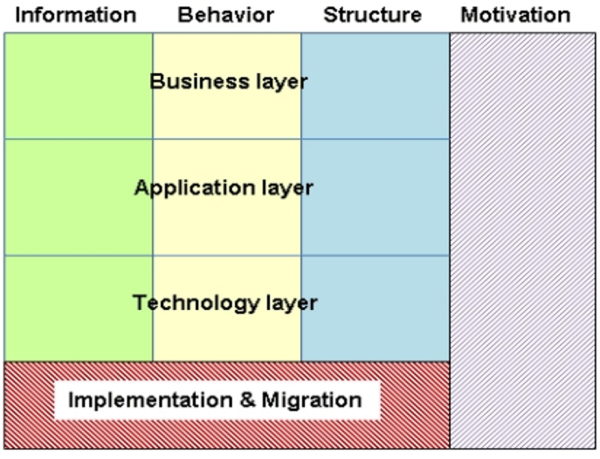


Figure 5.1: ArchiMate Framework, **Source:** ArchiMate® 2.0 specifications (The Open Group (2012))

ArchiMate is structured in three horizontal layers (Figure 5.1): the business layer, the appli-

cation layer and the technology layer. All three layers are built with the same type of concepts and the same sort of associations. They are structured according to three aspects (vertical layers). The first aspect regards the active structure elements which are defined as *entities that are capable of performing behaviour*, e.g., a role or an actor. The second aspect regards the behavioural elements which are defined as *units of activity performed by one or more active structure elements*, e.g., a process or a function. The last aspect addresses passive structure elements which are defined as *objects on which behaviour is performed*, e.g., a contract or an object.

Two types of extensions of ArchiMate are introduced in version 2.0: the Motivation extension (in grey in Figure 5.1) and the Implementation and Migration extension (in red). The first one is used to *model the motivations, or reasons, that underlay the design or change of some enterprise architecture* and the second one provides *concepts to support the implementation and migration of architectures.*

5.2.2 Core ArchiMate concepts

This section reviews the definitions of the core concepts from this business layer of ArchiMate. These definitions are extracted from the ArchiMate 2.0 specifications and are extended with additional explanation, when needed and if necessary for the analysis of the mapping in the next sections. Figure 5.2 represents the business layer core concepts of the ArchiMate metamodel modelled in UML.

- The concept of **value** of a product or service is defined *as the relative worth, utility, or importance of a business service or product.*
- The concept of **product** is defined in Jonkers et al. (2004) as *a collection of services, together with a contract that specifies the characteristics, rights and requirements associated with the product.* This definition is completed in Maria-Eugenia Iacob and Wiering (2004) by the idea that the product *is a coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.* This concept is associated to the concept of business service, of application service and, through the concept of contract, to the business object.
- The concept of **contract** is defined *as a formal or informal specification of an agreement that specifies the rights and obligations associated with a product.* The concept of contract is a type of business object.
- The concept of **business service** represents for Jonkers et al. (2004) *a unit of functionality that is meaningful from the point of view of the environment* and for Maria-Eugenia Iacob and Wiering (2004) *it represents the externally visible (logical) functionality, which is meaningful to the environment and is realised by business behaviour (business process, business function or business interaction).* Finally, the business service aims for the ArchiMate 2.0 specification, *to fulfil a business need for a customer (internal or external to the organisation).* This concept is realised by the concept of business process/function/collaboration which is directly assigned to business role(s).
- The concept of **business process** is defined as *a behaviour element that groups behaviour based on an ordering of activities. It is intended to produce a defined set of products or business services.*

- The concept of **business interface** is defined as *a point of access where a business service is made available to the environment.*
- The concept of **business collaboration** represents *an aggregate of roles within the organisation which performs collaborative behaviour.* This concept represents an aggregate of business roles assigned to a business interaction.
- The concept of **business interaction** represents *a behaviour element that describes the behaviour of a collaboration.*
- The concept of **location** is defined as *a conceptual point or extent in space* and is used to model the distribution of structural elements such as business actors, application components, and devices. This is modelled by means of an assignment relationship from location to structural element.
- The concept of **business object** is defined in Jonkers et al. (2004) as *the passive entities that are manipulated by behaviour such as business processes or function* and in Maria-Eugenia Iacob and Wiering (2004) as *a unit of information that has relevance from a business perspective.* The business object is accessed by the business event, is realised by representation, and is accessed by business function.
- The concept of **business function** is *a unit of internal behaviour that groups behaviour according to, for instance, required skills, knowledge, resources, and so forth, and is performed by a single role within the organisation.* For Jonkers et al. (2004), the business function *offers useful functionality that may be useful for one or more business processes.* The business function is assigned to a unique business role (this is a formal constraint from the ArchiMate specifications), it accesses business objects, it may trigger others business functions, and it triggers (or is triggered by) business events.
- The concept of **meaning** is defined *as the knowledge or expertise present in a business object or its representation, given a particular context.* The meaning represents the intention of a business object and is therefore associated to it.
- The concept of **representation** is defined as *a perceptible form of the information carried by a business object.*
- The concept of **business role** is defined as *a named specific behaviour of a business actor participating in a particular context.* Jonkers et al. (2004) additionally explain that *multiple actors can fulfil the same role, and conversely, a single actor can fulfil multiple roles.* The business role is assigned to a business function and to a business actor. In the ArchiMate v2.0 specifications, the business role is defined *as the responsibility for performing specific behaviour, to which an actor can be assigned.*
- The concept of **business actor** represents *an organisational active entity that is capable of performing behaviour.* This business actor can be an individual person (e.g., a customer or an employee) but also a group of people and resources that have a permanent (or at least a long–term) status within the organisation. This means that ArchiMate considers, for instance, a software agent, a department or a business unit as a business actor.
- The concept of **business event** is *something that happens (externally) and influences behaviour.* This business event triggers business processes, functions or interactions.

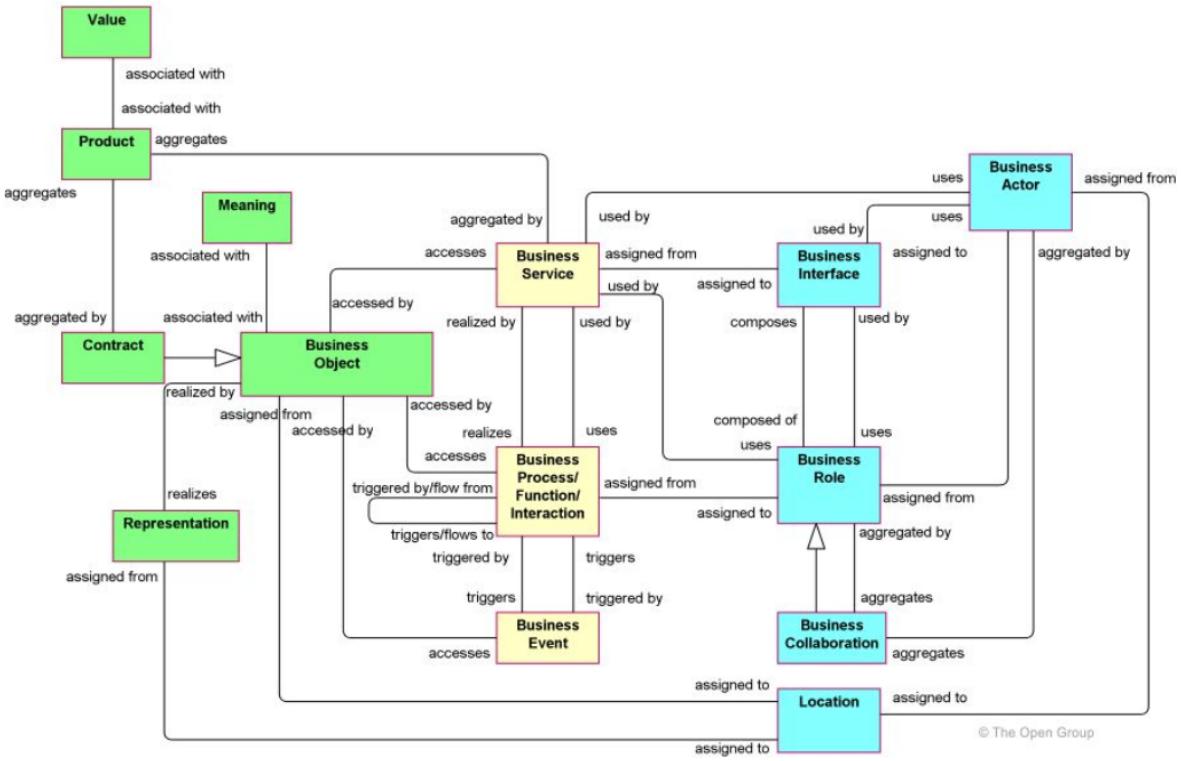


Figure 5.2: ArchiMate business layer, **Source:** ArchiMate® 2.0 specifications (The Open Group (2012))

All of these concepts are modelled in the metamodel shown on Figure 5.2. This metamodel does not specify all the existing types of relations between the concepts. A detail of these relations is provided in Appendix F, Figures F.1 and F.2. For the mapping, we need to consider the relations expressed in the metamodel as well as the ones provided in the appendix.

5.2.3 ArchiMate motivation extension

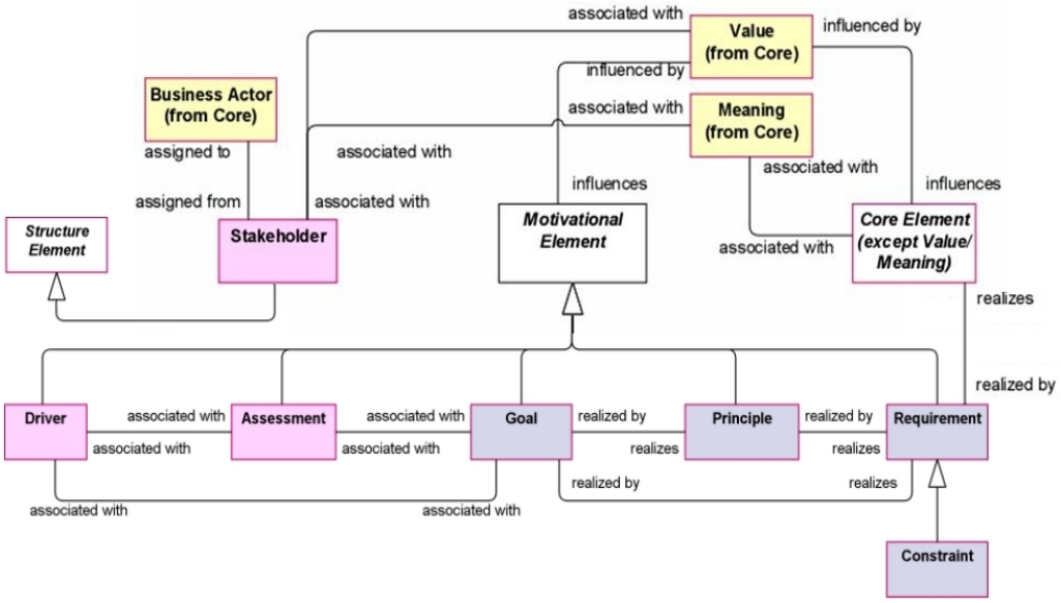


Figure 5.3: Relation between ArchiMate core concepts and the motivation concepts, **Adapted from:** ArchiMate® 2.0 specifications (The Open Group (2012))

Seven concepts make the motivation extension model (Figure 5.3):

- The concept of **stakeholder** is defined as *the role of an individual, team, or organisation (or classes thereof) that represents their interests in, or concerns relative to, the outcome of the architecture.*
- The concept of **driver** is defined as *something that creates, motivates, and fuels the change in an organisation.*
- The concept of **assessment** is defined as *the outcome of some analysis of some driver.*
- The concept of **goal** is defined as *an end state that a stakeholder intends to achieve.*
- The concept of **requirement** is defined as *a statement of need that must be realised by a system.*
- The concept of **constraint** is defined as a *restriction on the way in which a system is realised.*

- The concept of **principle** is defined as *a normative property of all systems in a given context, or the way in which they are realised.*

The ArchiMate motivation model allows expressing, as highlighted in Figure 5.3 that (1) a motivation element influences (*affects positively or negatively*) the value concept which is associated to a core concept of ArchiMate, and that (2) a motivation element of type requirement must be realised by a core concept. Additional relations between the concepts of the motivation extension are provided in Appendix F, Figure F.3.

5.2.4 ArchiMate modelling symbols

ArchiMate specifications v2.0 provides a specific symbol for each concept of the metamodel, thereby allowing the creation of models. The symbols of the concepts that we use in this chapter are represented in Figure 5.1 and correspond respectively to the business actor, the business function, the business object, the business role, the business process, the driver and the requirement. In the figure, we provide the concept name and the symbol of the instantiated concept. The instance of the concept is written in italic.

Equivalently, in Table 5.1, we provide the symbols of the six associations between concepts that we also need in the following. These concepts are the association, the aggregation, the assignment, the read access, the write access, and the read–write access.

5.2.5 ArchiMate extension mechanisms

ArchiMate core and motivation concepts and relations between concepts may be extended using two extension mechanisms.

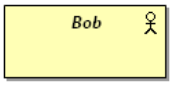
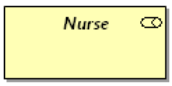
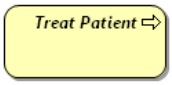
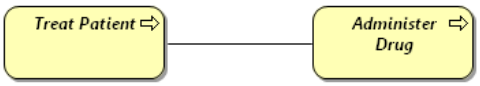
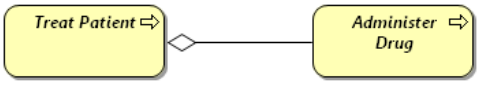
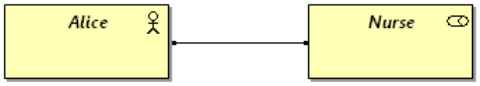
The first extension mechanism consists in adding supplementary information to the elements (concepts and relations) of the metamodel. The addition of information is realised by defining attributes to the existing ArchiMate element. These attributes consist of a name and a type. For instance, `Class 1` is an ArchiMate concept modelled in UML (Figure 5.4(a)), which is extended by the attribute `attributeName1` (Figure 5.4(b)).

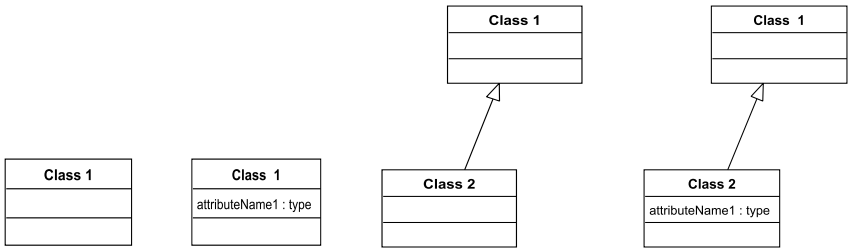
The second extension mechanism consists in specialising elements using stereotypes. These specialised elements inherit the properties of their parents elements including additional restrictions. ArchiMate specifications point out that the specialisation *strongly resembles a stereotype as it is used in UML*. From a modelling point of view, `Class 2` is an ArchiMate concept modelled in UML which is an extension of `Class 1`, as represented in Figure 5.4(c).

Figure 5.4(d) represents the cumulation of both types of extension mechanism.

ArchiMate specifications v2.0 considers that the two extensions also apply to the relation between concepts. This extension may be modelled using UML. For instance, the extension of the `Association class 1` in Figure 5.5(a) with an attribute is illustrated in Figure 5.5(b) and with an attribute and a stereotype is illustrated in Figure 5.5(c).

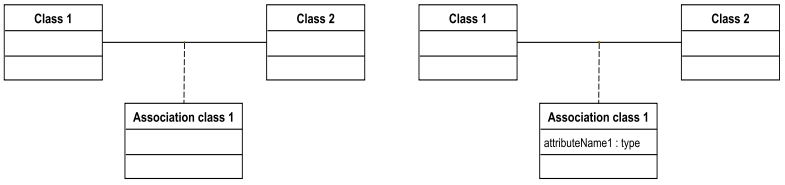
Table 5.1: ArchiMate concepts and associations between concept's symbols, **Source:** ArchiMate® 2.0 specifications (The Open Group (2012))

Concept name	Concept symbol
Business actor:	
Business function:	
Business object:	
Business role:	
Business process:	
Driver:	
Requirement:	
Association:	
Aggregation:	
Assignment:	
Read:	
Write:	
Read–write:	

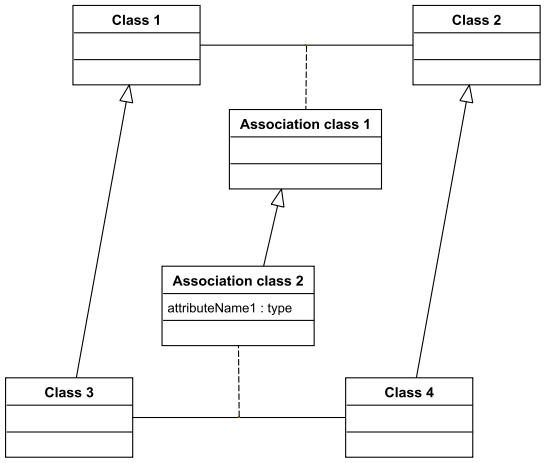


(a) UML Class (b) UML Class with attribute (c) UML Class specialisation (d) UML Class specialisation with attribute

Figure 5.4: Class extension mechanisms



(a) UML Association Class (b) UML Association Class with attribute



(c) UML Association Class specialisation

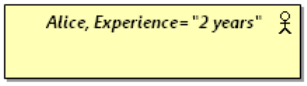
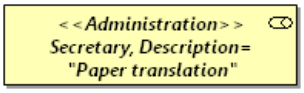
Figure 5.5: Relation between classes extension mechanisms

5.2.6 ArchiMate motivation extension modelling symbols

With regards to the new concepts and relations engineered using the extension, no formalism for representing the attributes and/or stereotypes is imposed by the ArchiMate specifications. For the sake of clarity, we propose our own representation of them based on the existing ArchiMate

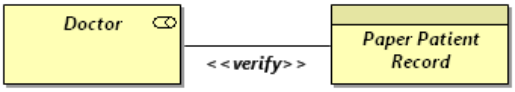
symbols. Concerning the attribute (such as modelled in UML in Figure 5.4(b)), we list the element's attributes after the name of this element. E.g., the ArchiMate business actor of Table 5.2 is extended with the attribute *Experience*. An instance of this extended concept is *Alice* which as an *Experience*=*"2 years"*. Concerning the stereotype (like modelled in UML in Figure 5.4(c)), we write the name of the stereotype between quotation marks as it is usually performed in UML[1]. This is represented in Table 5.2 by the *Secretary* which is an instance of the ≪*Administration*≫ which is a stereotype of business role. Equivalently, the cumulation of both extension mechanisms (modelled in UML in Figure 5.4(d)) is illustrated in the table by the *Secretary* which is a instance of the ≪*Administration*≫ stereotype of business role and which has the attribute *Description* of value *"Paper translation"*.

Table 5.2: ArchiMate concept extension symbols

ArchiMate concept extension	Concept extension symbol
ArchiMate concept with attribute:	
ArchiMate concept specialisation:	
ArchiMate concept specialisation with attribute:	

With regard to the relation, when a stereotype extends an ArchiMate relation between concepts (such as modelled in UML in Table 5.3), the name of the stereotype is written between quotation marks. For example, the verify relation between a business role and business object is a stereotype of the association relation. This is represented by labelling the association with ≪*verify*≫. No case of association extension with an attribute has been encountered in our mapping. Hence no representation of this extension mechanism is provided.

Table 5.3: ArchiMate association extension symbol

ArchiMate association extension	Association extension symbol
ArchiMate association specialisation:	

[1]In UML, *the name of the stereotype is shown within a pair of guillemots above or before the name of the model element.*

5.3 Resolution of heterogeneities during metmodels integration

The mapping between ArchiMate and the Responsibility metamodel aims at extending the EAM with a responsibility perspective. In order to integrate two metamodels, Zivkovic et al. (2007) explain that three types of heterogeneity need to be resolved: semantic, structural, and syntactic.

5.3.1 Semantic heterogeneity

According to Zivkovic et al. (2007), the semantic heterogeneity represents differences in the meaning of the considered metamodels elements and must be addressed through elements mapping and integration rules. The elements mapping introduces a correspondence between at least one element of each of the source metamodels. According to Parent and Spaccapietra (2000), two types of mapping are conceivable: 1:1 and n:m mapping. A 1:1 mapping means a correspondence between two elements of two sets of objects (from two different models) which corresponds to the *equivalence* between elements from Zivkovic et al. In our mapping, the integration rule for these elements is a *merge* into a unique element in the target metamodel, and all the attributes of the source elements are assigned to this unique element. One source element may be semantically richer/poorer that the other elements, e.g., be more general or more specific, the mapping between the two elements exists with, respectively, a generalisation/specialisation conflict (according to Parent and Spaccapietra (2000)). In this case, both concepts are associated in the integrated metamodel with a generalisation/specialisation relationship. This matches the correspondence of a type *relation* from Zivkovic et al.

The mapping of a type n:m relates to a set of elements from one metamodel to a set of elements from the other so that no 1:1 mapping between the elements of the two sets exist. This second type of mapping exists when the mapping requires the resolution of fragmentation conflicts (conflicts which arise from a different decomposition of the real world elements being modelled – Parent and Spaccapietra (2000)). No occurrence of this n:m mapping has been encountered amongst the ArchiMate and ReMMo and, as a result, this mapping will be no further explained here.

If no mapping exist between two elements from the analysed metamodels, we are in the case of *non–relation* correspondence described by Zivkovic et al. In this case, both elements from the analysed metamodels need to be represented in the integrated metamodel as, e.g., a concept, an attribute, an association. In our integration of the Responsibility metamodel with ArchiMate, when no mapping exist, the element which only exists in the Responsibility metamodel will be represented in the integrated metamodel using the ArchiMate extension mechanism which consists in adding attribute to an existing elements.

5.3.2 Structural heterogeneity

The structural heterogeneity exists when the same metamodel concepts are modelled differently by each metamodel primitives. For instance, when a concept is represented by a class in one metamodel and is a represented by a relation in another metamodel, or when a concept is represented by a class in the first metamodel or by two classes in the second. This structural heterogeneity will be addressed together with the analysis of the conceptual mapping and the definition of the integration rules in Section 5.4.

5.3.3 Syntactic heterogeneity

This last type of heterogeneity is not relevant to us. Indeed, the syntactic heterogeneity aims at analysing the difference between the serialisation of metamodel and, as explained by Busse et al. (1999), addresses technical heterogeneity like hardware platforms and operating systems, or access methods, or it addresses the interface heterogeneity like the one which exists if different components are accessible through different access languages. Similarly, Parent and Spaccapietra (2000) considers that the syntactic heterogeneity is the most visible type of heterogeneity and that it must be addressed by performing a syntactic rewriting during the preparation step of the integration of two databases. Regarding our mapping, this syntactic heterogeneity is not applicable since no serialisation format for storing the Responsibility metamodel has been provided until now. Only the semantic and structural heterogeneities are therefore considered, and relevant, in our case.

5.4 Mapping between ArchiMate and the Responsibility metamodel

To perform the conceptual mapping between the two metamodels, we have systematically analysed each concept of ReMMo to better understand to which concept of the ArchiMate core and motivation extension they correspond and we have resolved heterogeneity issues such as explained in Section 5.3.

Afterwards, we have systematically analysed and resolved heterogeneity issues regarding associations between the concepts from both metamodels. To that end, the associations between concepts from the Responsibility metamodel have been modelled using the *association classes* and the latter have been named based on the semantic of the associations. These association classes are presented in Figure 5.6. For instance, the association class between *Task* and *BusinessObject* which has the semantic: *the task uses the business object* and inversely, *the business object is used by the task* is named *Use association*.

To be exhaustive, the mapping has also considered the heterogeneity regarding the cardinalities and the constraints related to the latter. In ArchiMate, these cardinalities are assumed to be zero to many, unless being explicitly shown in the metamodel, and no constrains have been provided. In the Responsibility metamodel, cardinalities are defined for each associations, and the six constraints regarding the latter are written in natural language. During the integration of the Responsibility metamodel with ArchiMate, we set, one for all, that the cardinalities related to ReMMo are always more constraining than the ones from ArchiMate, with regards to the purpose of our integration. Therefore, the cardinalities and constraints from the Responsibility metamodel are always going to be those applying in the integrated metamodel.

In the remainder of this section, we firstly analyse the mapping for the task, business task, structural task, approve task and business object in Section 5.4.3, then, the business role, the employee, the responsibility, the accountability and the right to use in Section 5.4.4, the condition, sanction and capability in Section 5.4.5 and the source and governance rules in 5.4.6.

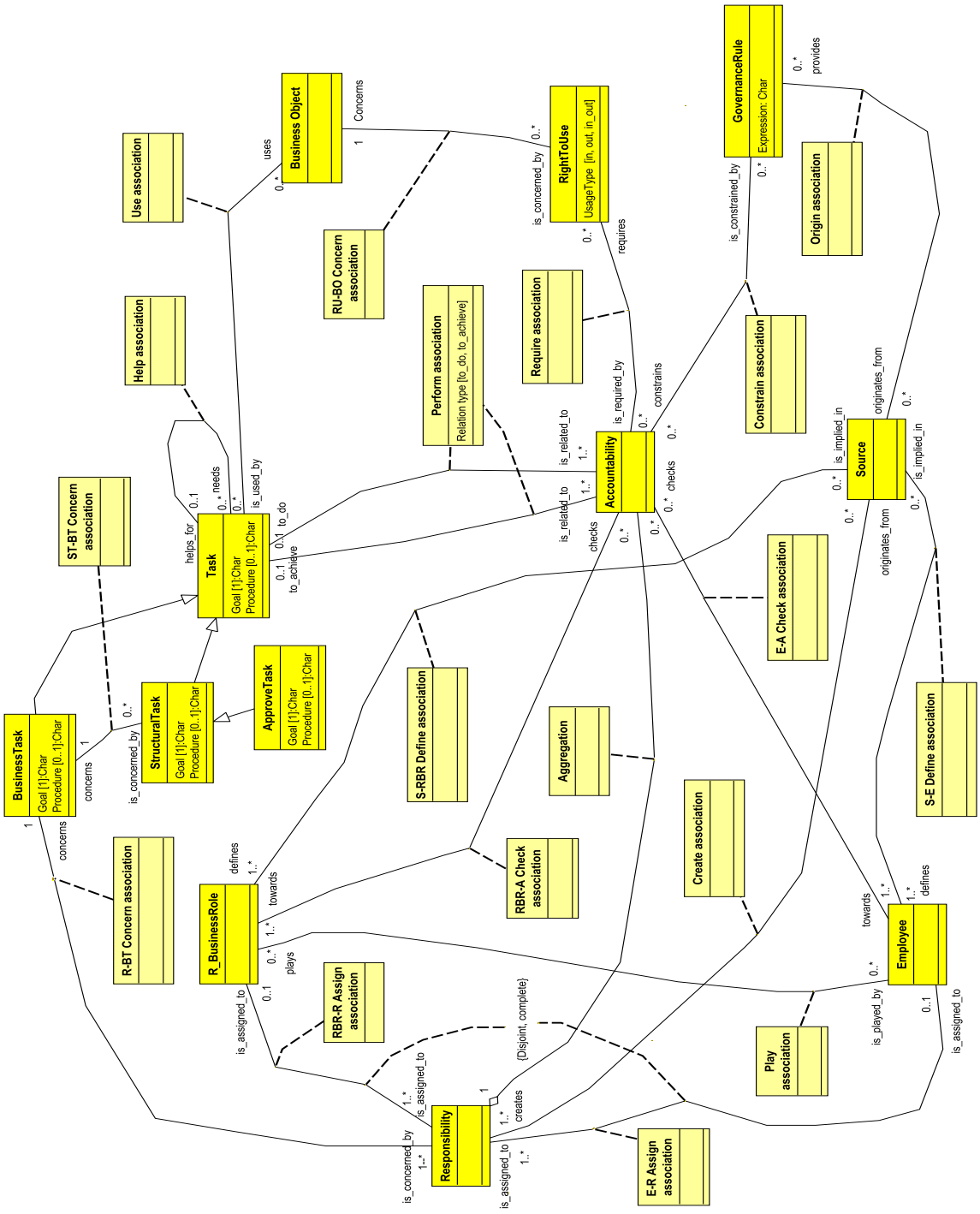


Figure 5.6: Responsibility metamodel association classes

5.4.1 ArchiMate metamodel UML fragment

To perform the mapping, we have also modelled the concepts of the ArchiMate business layer concerned by the integration in an UML schema. In this schema, all the cardinalities have been represented, as well as the associations between concepts that have been modelled using association class. For instance, the association between the business process (BP) and the business function (BF) is an aggregation modelled by the class `BP-BF Aggregation`. This UML schema of the ArchiMate metamodel is represented on Figure 5.7.

5.4.2 Graphical convention for our Responsibility ArchiMate extension

In the following figures of this chapter, which represented in UML the integration of ReMMo with ArchiMate, the classes in white correspond to ArchiMate concepts which have no counterpart in the Responsibility metamodel, the classes in grey correspond to original ArchiMate associations, the classes in dark yellow correspond to concepts originating from the Responsibility metamodel and the classes in light yellow correspond to associations between concepts originating from this Responsability metamodel. The classes in green represent original ArchiMate concepts or associations which have an equivalent in ReMMo.

5.4.3 Task, Business Task, Structural Task, Approve Task, Business Object and Right to Use

Figure 5.8 represents the result of the integration between the task, the business task, the structural task, the approve task, the business object and the right to use from the Responsibility metamodel, and the business process and the business object from ArchiMate.

The semantic of the business object from ArchiMate corresponds to the semantic of the business object from the Responsibility metamodel. In ArchiMate, the business object is associated with a meaning and is realised by a representation. In ReMMo, the meaning and the representation are not explicit. Despite the non–existence of such an association in the Responsibility metamodel, we consider a 1:1 mapping without conflict between the business object from ArchiMate and the business object from the Responsibility metamodel. Therefore, the integration rule is *merge* and both concepts are merged in a common one named Business Object and represented by the class `Business Object` in Figure 5.8.

In ReMMo, we have defined the task as *a complete and identifiable piece of work necessary to achieve a goal and which may or may not be defined with a procedure*. This task has one goal, and zero or one procedure as attributes. According to the definition of the business process from ArchiMate, we consider that the concept of task is more specific than the concept of business process. The correspondence between the two concepts is a 1:1 mapping and the integration rule between the two concepts is a *specialisation* such that the task is a specialisation of the business process. To integrate both concepts, we apply the stereotype extension mechanism defined in Figure 5.4(c). This is afterwards expressed as the ≪`Task`≫ as a stereotype of the `Business Process`.

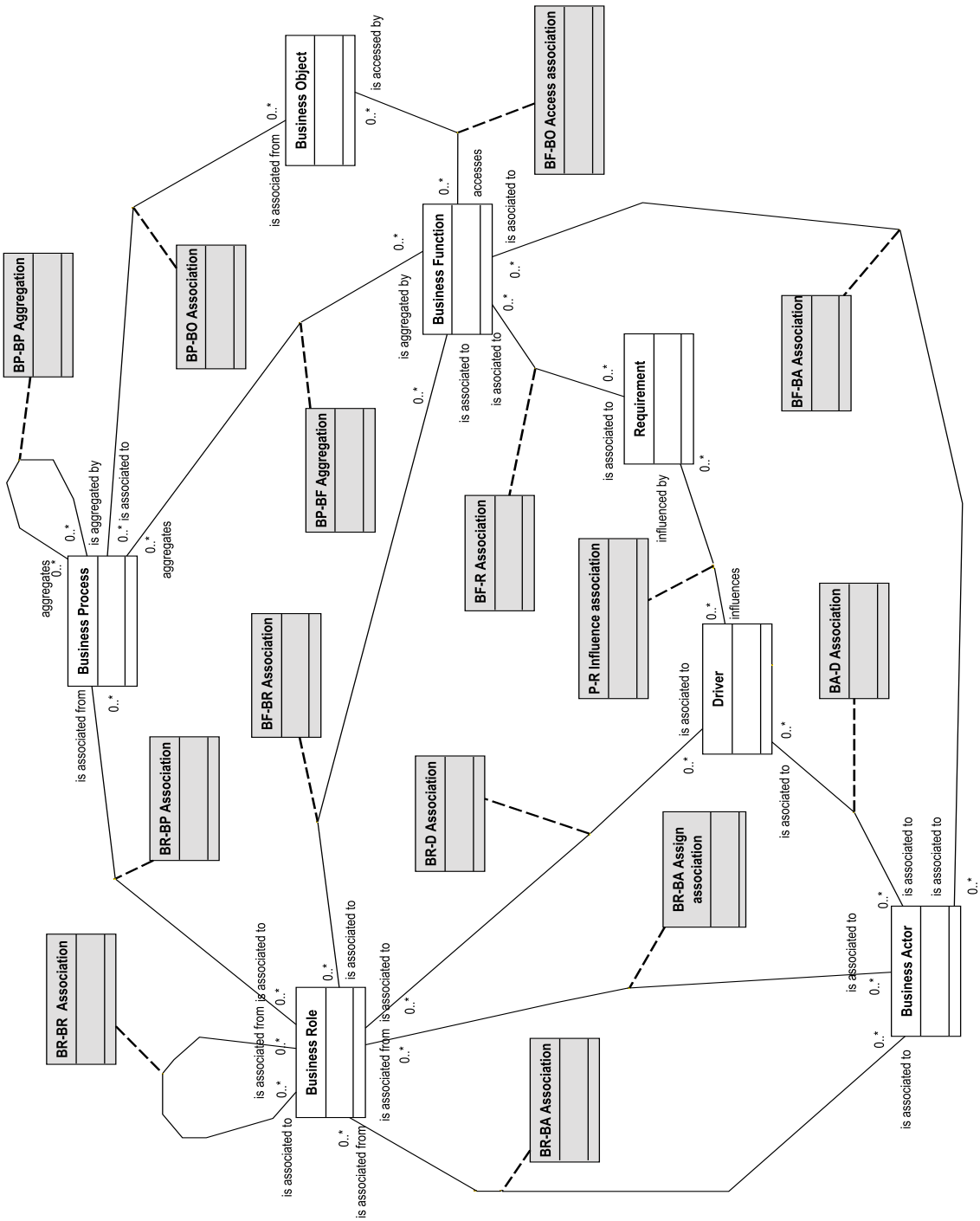


Figure 5.7: ArchiMate metamodel UML fragment

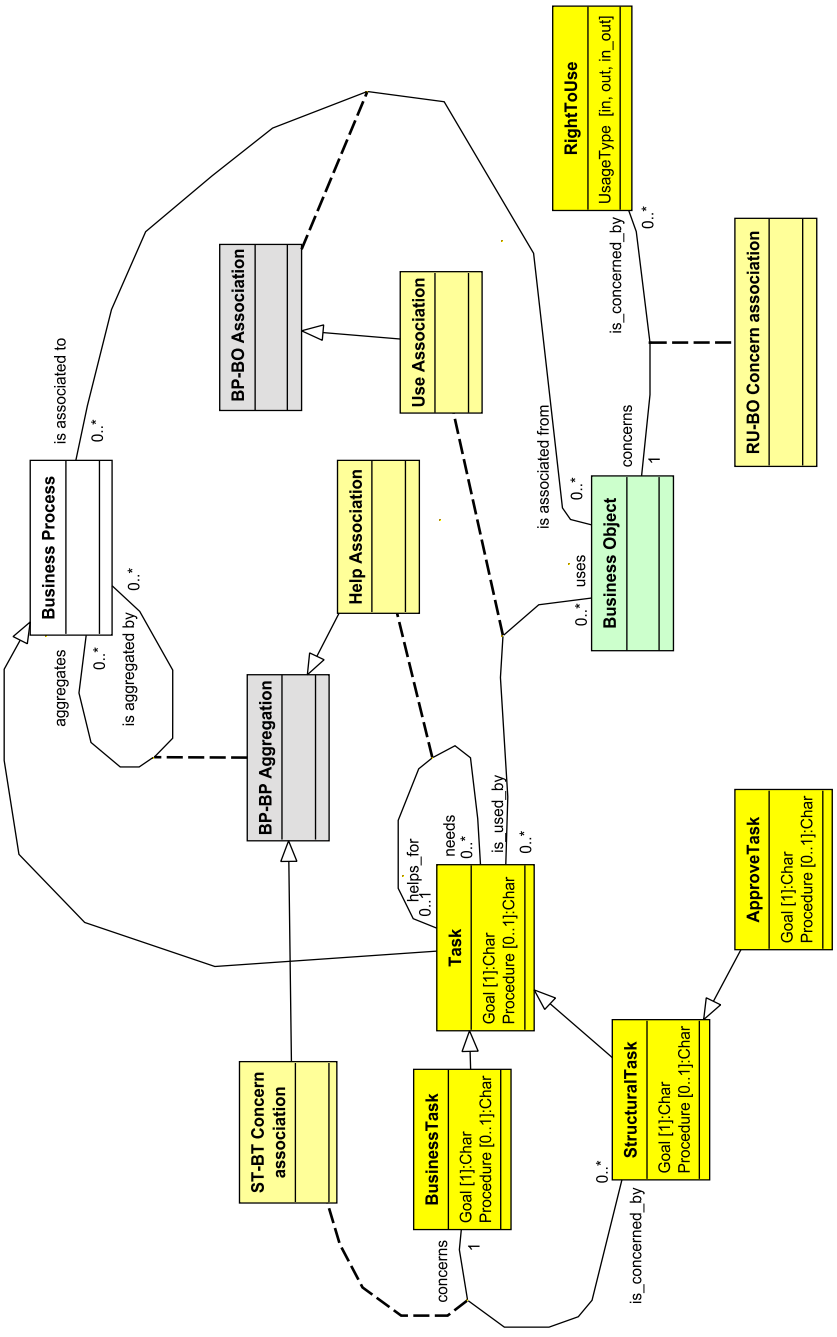


Figure 5.8: Task, Business task, Structural task and Business object conceptual mapping

The business task and the structural task from the Responsibility metamodel are both types of tasks. Therefore, in the integrated meatmodel, we create two sub–classes of the class `Task` which we name `BusinessTask` and `StructuralTask`. Similarly, approve, supervise, support, advise, control and report tasks are types of structural tasks. Hence, we create sub–classes of the class `StructuralTask` which we name, respectively, `ApproveTask`, `SuperviseTask`, `SupportTask`, `AdviseTask`, `ControlTask` and `ReportTask`. To keep the Figure 5.8 readable, only the `ApproveTask` is represented and the other types of structural tasks are dealt with equivalently.

Regarding the relations, in the Responsibility metamodel, a task may be associated to another task through the *Help association*. In ArchiMate, a business process may aggregate others business processes. Both associations express the decomposition of tasks into sub–tasks. We consider that there exists a 1:1 mapping between the *Help association* and the *BP-BP Aggregation* from ArchiMate. Given that this *Help association* expresses the reason why a task is associated to another task, we consider that this *Help* association is more specific than the *BP-BP Aggregation*. To integrate both, we apply the stereotype extension mechanism defined in Figure 5.5(c) so that the *Help association* from the Responsibility metamodel specialises the *BP-BP Aggregation* from ArchiMate. The ArchiMate graphical syntax for representing this aggregation relation is illustrated in Figure 5.9 where `Business process 1 aggregates Business Process 2`.

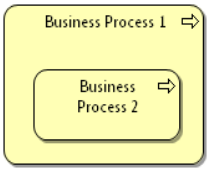


Figure 5.9: Business Process 2 is aggregated with Business Process 1

Within the same reasoning, we consider that there exists a 1:1 mapping between the *ST-BT Concern association* between a structural task and a business task from ReMMo, and the *BP-BP Aggregation* between business processes from ArchiMate. Equally, as the association from the Responsibility metamodel is semantically more specific than the association from ArchiMate, we apply the stereotype extension mechanism such that the *ST-BT Concern association* from the Responsibility metamodel specialises the *BP-BP Aggregation* from ArchiMate.

Finally, in the Responsibility metamodel, we have expressed through the *Use association* that a task is associated to a business object but does not represent the type of rights related to the usage of the business object (i.e., read, write, read–write). Therefore, in the integrated metamodel, we consider that there exists a 1:1 mapping between the *Use association* which relates a task and a business object from ReMMo and the *BP-BO Association* between the business process and the business object from ArchiMate. As the association from ArchiMate is semantically more generic as the *Use association* from the Responsibility metamodel, we consider that the *Use association* specialises the *BP-BO Association* from ArchiMate. Additionally, in order to represent the right to use the business object, we have created in the Responsibility metamodel a class `RighToUse` which is associated to the business object by the *RU-BO Concern*

association, and which is part of the integrated metamodel, as shown on Figure 5.8.

5.4.4 Business Role, Employee, Responsibility, Accountability

In ReMMo, the business role has been defined as *a type of actor which represents a set of employees who share common characteristics and who are assigned to the same responsibilities.* In ArchiMate, the definition of the business role has changed with the evolution of the ArchiMate specifications. Before version 2.0, the definition of the business role focused on *a named specific behaviour.* In version 2.0, the business role is defined by *the responsibility for performing specific behaviour, to which an actor can be assigned* but in this version of the specification, it is also explained that *business processes or business functions are assigned to a single business role with certain responsibilities or skills.* This means that the business role is not considered as a responsibility in itself but that it is an element that possesses responsibilities. ArchiMate also explains that a business role is useful in a (structural) organisational sense such as in the division of labour within an organisation, and that it corresponds, for instance, to an Insurance Seller or an Insurance Buyer (Jonkers et al. (2012)). Finally, the business layer metamodel (Figure 5.2) represents the business actors and the business roles as active entities (subjects) who perform behaviour such as the business functions.

Metamodel	**ArchiMate**		**Responsibility**		
Concept / Meaning	Business actor	Business role	Employee	Business role	Responsibility
One human only	**Yes**	No	**Yes**	No	No
A set of humans	**Yes**	**Yes**	No	**Yes**	No
Responsibility	No	**Yes**	No	No	**Yes**

Table 5.4: Meaning comparison between Business actor, Business role, Employee, Responsibility

As summarised in Table 5.4, the meaning of the business role in ArchiMate seems ambiguous as it signifies a subject that performs a behaviour (explanations of the business layer metamodel) or a responsibility to perform a specific behaviour (definition of the concept). Firstly, we observe that there exists a correspondence between the business role from ArchiMate and the business role from the Responsibility metamodel. Therefore, we apply the stereotype extension mechanism and we associate both business roles using the specialisation relation so that the business role from the Responsibility metamodel is a specialisation of the business role from ArchiMate. This integration is represented in Figure 5.10 and is expressed by the ≪`R_BusinessRole`≫ as a stereotype of the `Business Role` from ArchiMate. Not to introduce confusion with the business role from ArchiMate and not to have two concepts with the same name, this stereotype is named **R**_BusinessRole to signify that it is originating from the **R**esponsibility metamodel.

Secondly, the responsibility has been defined in the Responsibility metamodel as *a charge assigned to a unique actor to signify its accountabilities concerning a unique business task.* Hence, we also observe a correspondence between the business role from ArchiMate and the responsibility from the Responsibility metamodel (Table 5.4). Therefore, we apply the stereotype extension

mechanism and we associate both using the specialisation relation so that the responsibility from the Responsibility metamodel is a specialisation of the business role from ArchiMate. This is expressed by the ≪`Responsibility`≫ as a stereotype of the `Business Role`.

In ReMMo, the employee has been defined by *a type of actor which represents a human entity which may or may not play, one or more business roles.* Regarding ArchiMate, as explained in Section 5.2.2, the business actor may be of the type: *individual person* or *group of people, who have a permanent (or at least a long–term) status within the organisation.* Accordingly, we observe the correspondence between the business actor from ArchiMate and the employee from the Responsibility metamodel. Therefore, we apply the stereotype extension mechanism and we associate both using the specialisation relation so that the employee from ReMMo is a specialisation of the business actor from ArchiMate. This mapping is represented in Figure 5.10. It is expressed by the ≪`Employee`≫ as a stereotype of the `Business Actor`.

The accountability is defined in the Responsibility metamodel as an element which composes a unique responsibility and which represents *an obligation of an actor to achieve a goal, or to perform the procedure of a task, and the justification that it is done to someone else, under threat of sanction.* In the Responsibility metamodel, this accountability is related to a task by a *to do* or a *to achieve* association, is associated to a sanction, requires capability and rights to use, and is due toward another employee/business role. In ArchiMate, the business function is defined as *a behaviour element that groups behaviour based on a chosen set of criteria (typically required business resources and/or competences).* We observe the correspondence between the business function from ArchiMate and the accountability. However, in ArchiMate, the business function may be associated to one or more business processes although in ReMMo, the accountability is related to only one task. As a consequence, we apply the stereotype extension mechanism and we associate them using a specialisation relation such as the accountability from the Responsibility metamodel is a specialisation of the business function from ArchiMate. This is expressed by the ≪`Accountability`≫ as a stereotype of the `Business Function`.

Regarding the relations, in the Responsibility metamodel, a responsibility aggregates an accountability. In ArchiMate, the business role may not be modelled as an aggregation of business functions, and, consequently, the ≪`Responsibility`≫ stereotype of the `Business Role` may not be modelled as an aggregation of the ≪`Accountability`≫ stereotype of the `Business Function`. As a result, the *BF-BR Association* relation is used to model the association between both concepts and we observe a 1:1 mapping between the *BF-BR Association* from ArchiMate and the *Aggregation* from ReMMo. Therefore, we apply the stereotype extension mechanism and the association between both association classes is a specialisation such that the *Aggregation* from the Responsibility metamodel is an ≪`Aggregation`≫ stereotype of the `BF-BR Association`. The ArchiMate graphical syntax for representing this aggregation relation is illustrated in Figure 5.11 where `Business process 1 aggregates Business Function 1`.

The responsibility concerns a business task. This is represented by the `R-BT Concern association` class. In ArchiMate, the business role is associated to the business process. This is represented by the `BR-BP Association` class. We observe a mapping 1:1 without conflict between both types of association. Therefore, the integration rule is *merge* and both association classes are merged in a common one named `BR-BP Association`.

In the Responsibility metamodel, the employee is related to the business role through the

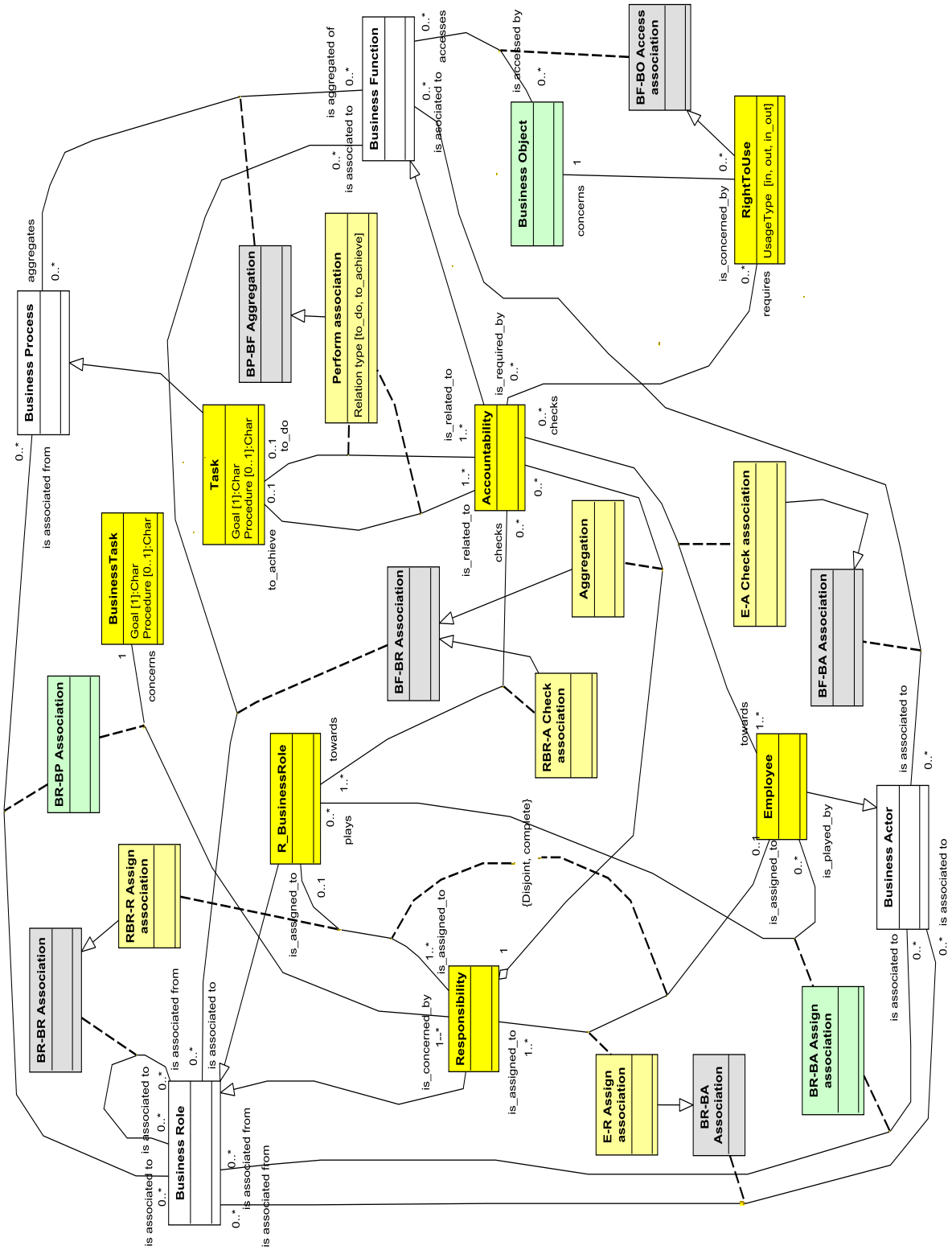


Figure 5.10: Business role, Employee, Responsibility, Accountability conceptual mapping

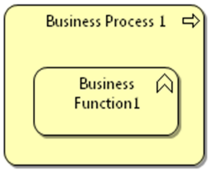


Figure 5.11: Business Function 1 is aggregated with Business Process 1

Play association. In ArchiMate, the business actor is assigned to the business role. This is represented by the `BR-BA Assign association` class. We observe a mapping 1:1 without conflict between both types of association. Therefore, the integration rule is *merge* and both association classes are merged in a common one named `BR-BA Assign association`.

The accountability from the Responsibility metamodel is, firstly, possibly checked by an employee. This is represented by the *E-A Check association*. In ArchiMate, the business function is associated to the business actor. This is represented by the `BF-BA Association` class. We observe a 1:1 mapping between the *BF-BA Association* from ArchiMate and the *E-A Check association* from the Responsibility metamodel. As the *E-A Check association* is more specific than the *BF-BA Association*, we apply the stereotype extension mechanism and the association between both association classes is a specialisation so that the *E-A Check association* from the Responsibility metamodel is an ≪`E-A Check`≫ stereotype of the `BF-BA Association` class from ArchiMate. Secondly, the accountability is possibly checked by a business role. This is represented by the *RBR-A Check association*. We equivalently observe a 1:1 mapping between the *BF-BR Association* from ArchiMate and the *RBR-A Check association* from ReMMo. As the *RBR-A Check association* is more specific than the *BF-BR Association*, we also apply the stereotype extension mechanism. The association between both association classes is a specialisation so that the *RBR-A Check association* from the Responsibility metamodel is an ≪`RBR-A Check`≫ stereotype of the `BF-BR Association` class from ArchiMate.

The accountability from the Responsibility metamodel is to_do or to_achieve a task. This is represented by the `Perform association` class. In ArchiMate, the business function is aggregated with the business process. This is represented by the `BP-BF Aggregation` class. We observe a mapping 1:1 between both types of relation. Therefore, we apply the stereotype extension mechanism and the association between both association classes is a specialisation so that the `Perform association` class from ReMMo is an ≪`Perform association`≫ stereotype of the `BP-BF Aggregation` class from ArchiMate which has the attribute `Relation type: to_do or to_achieve` from the Responsibility metamodel.

In the Responsibility metamodel, the responsibility is assigned to an employee or to a business role. This is represented, respectively, by the `E-R Assign association` and the `RBR-R Assign association` classes. In ArchiMate, the business role is associated to the business actor or to itself. This is represented, respectively, by the `BR-BA Association` and the `BR-BR Association` classes. We observe a mapping 1:1 without conflict between the `E-R Assign association` and the `BR-BA Association` classes, and between the `RBR-R Assign association` and the `BR-BR Association` classes. Therefore, the integration rule is *merge* and (1) the `E-R Assign association` and the `BR-BA Association` classes are merged in a common one named `BR-BA`

`Association` and (2) the `RBR-R Assign association` and the `BR-BR Association` classes are merged in a common one named `BR-BR Association`.

Finally, the Responsibility metamodel allows representing that an accountability requires rights to use. This right to use has been defined as *an authorisation to perform an operation on a business object and which is required to perform an accountability.* We observe the mapping 1:1 between the right to use concept from ReMMo and the *BF-BO Access association* from ArchiMate. Therefore, we apply the stereotype extension mechanism and the association between both is a specialisation so that the `RightToUse` class from the Responsibility metamodel is an ≪`RightToUse`≫ stereotype of the `BF-BO Access association` class from ArchiMate. The ArchiMate graphical syntax for representing this access relation is a dash line with an arrow which indicate the sense of the information flow and, hence, if the access is of a type read, write or read–write.

5.4.5 Condition, Sanction and Capability

It has not been possible, due to semantic heterogeneity, to map the concepts of condition, sanction and capability with another existing concept of ArchiMate. This corresponds to the situation of non–relation depicted by Zivkovic et al. (2007). As mentioned in Section 5.3.1, if no mapping exists, the concepts may be represented as attributes of other concepts. This extension of the ArchiMate concept with attributes corresponds to the mechanism illustrated in Figure 5.4(d). In our case, we consider the capability, the condition and the sanction as attributes of the accountability, which is justified by the fact that the sanction results from the justification of the realisation (or not) of one accountability, that this accountability exists under one or more condition(s) and finally, that this accountability also requires capability.

Concerning the capability, we also note that a movement currently exists regarding the integration of the latter within the enterprise architecture frameworks, including ArchiMate. Among the ongoing works involved in this progress, the white paper van Dijk et al. (2013) analyses the enterprise architecture capability and defines it as *the business function that includes the people, processes and technology which are needed to execute enterprise architecture processes, and deliver enterprise architecture products.* TOGAF proposes the Architecture Capability Framework[1] and argues that to operate an architecture function within an enterprise, it is necessary to *put in place appropriate organisation structures, processes, roles, responsibilities, and skills to realise the architecture capability.* This architecture capability must be distinguished from the business capability like addressed by Iacob et al. (2012) who define a Valuation Extension to ArchiMate. This value extension proposes the capability as a new concept and defines it as *the ability (of a static structure element, e.g., actor, application component, etc.) to employ (configure, integrate, etc.) resources to achieve goals.* Vicente et al. (2013) analyse the relation between enterprise architecture and ITIL (ITIL (2001)) and represents the latter processes using enterprise architecture. Therefore, they analyse the value, and more specifically the capability, of these processes on the basis of the valuation extension proposed by Iacob et al. (2012).

[1] http://pubs.opengroup.org/architecture/togaf9-doc/arch/

5.4.6 Source and Governance Rules

Source and Governance rules are concepts from the Responsibility metamodel which are at an higher abstraction layer than the other concepts since they aim representing the elements that motivate the definition of the responsibilities. Therefore, the mapping between these concepts and the motivation extension concepts have been analysed, as illustrated in Figure 5.12.

The source is defined as *a formal piece of information which creates responsibilities and which contains, amongst others, required or desired governance rules*. We observe a mapping 1:1 between the source and the motivation concept of driver. Therefore, we apply the stereotype extension mechanism and the association between both classes is a specialisation so that the `Source` class from the Responsibility metamodel is a specialisation of the driver from ArchiMate. This is represented by the ≪`Source`≫ is a stereotype of the `Driver`.

The governance rule is defined as *a high level prescript originating from dedicated sources and which constrains the definition of the accountabilities*. We observe a mapping 1:1 between the governance rule and the motivation concept of requirements. Therefore, we apply the stereotype extension mechanism and the association between both classes is a specialisation so that the `Governance rule` class from the Responsibility metamodel is a specialisation of the `Requirement` class from ArchiMate. This is represented by the an ≪`Governance rule`≫ as a stereotype of the `Requirement` from ArchiMate. Moreover, this ≪`Governance rule`≫ has the attribute `Expression: Expression of the governance rule`.

Regarding relations, in ReMMo, the governance rule is associated to the source with the *Origin association* although in the motivation extension model, a driver is associated to a requirement by the *D-R Influence association* (Figure 5.3). Therefore, we consider that a 1:1 mapping exists between the *Origin association* from the Responsibility metamodel and the *D-R Influence association* from ArchiMate and, as the *Origin association* is semantically more specific than the *D-R Influence association*, we apply the stereotype extension mechanism and the association between both classes is a specialisation so that the `Origin association` class from the Responsibility metamodel is a specialisation of the `D-R Influence association` class from ArchiMate. This is represented by ≪`Origin association`≫ as a stereotype of the `D-R Influence association` class from ArchiMate.

In the Responsibility metamodel, the governance rule is related to the accountability with a `Constrain association` class. In ArchiMate, the requirement is related to the business function with the `BF-R Association` class. We observe a mapping 1:1 between the both association classes and therefore, we apply the stereotype extension mechanism. The association between both classes is a specialisation so that the `Constrain association` is a specialisation of the `BF-R Association` class from ArchiMate. This is represented by the ≪`Constrain association`≫ as a stereotype of the `BF-R Association` class from ArchiMate.

Equivalently, the source is related to the business role and to the responsibility with, respectively, a `S-BRB Define association` and a `Create association` classes. In ArchiMate, the driver is related to the business role with the `BR-D Association` class. We observe a mapping 1:1 between both association classes from ReMMo and the association class from ArchiMate. Therefore, we apply the stereotype extension mechanism. The association between the association classes from the Responsibility metamodel and the association class from ArchiMate

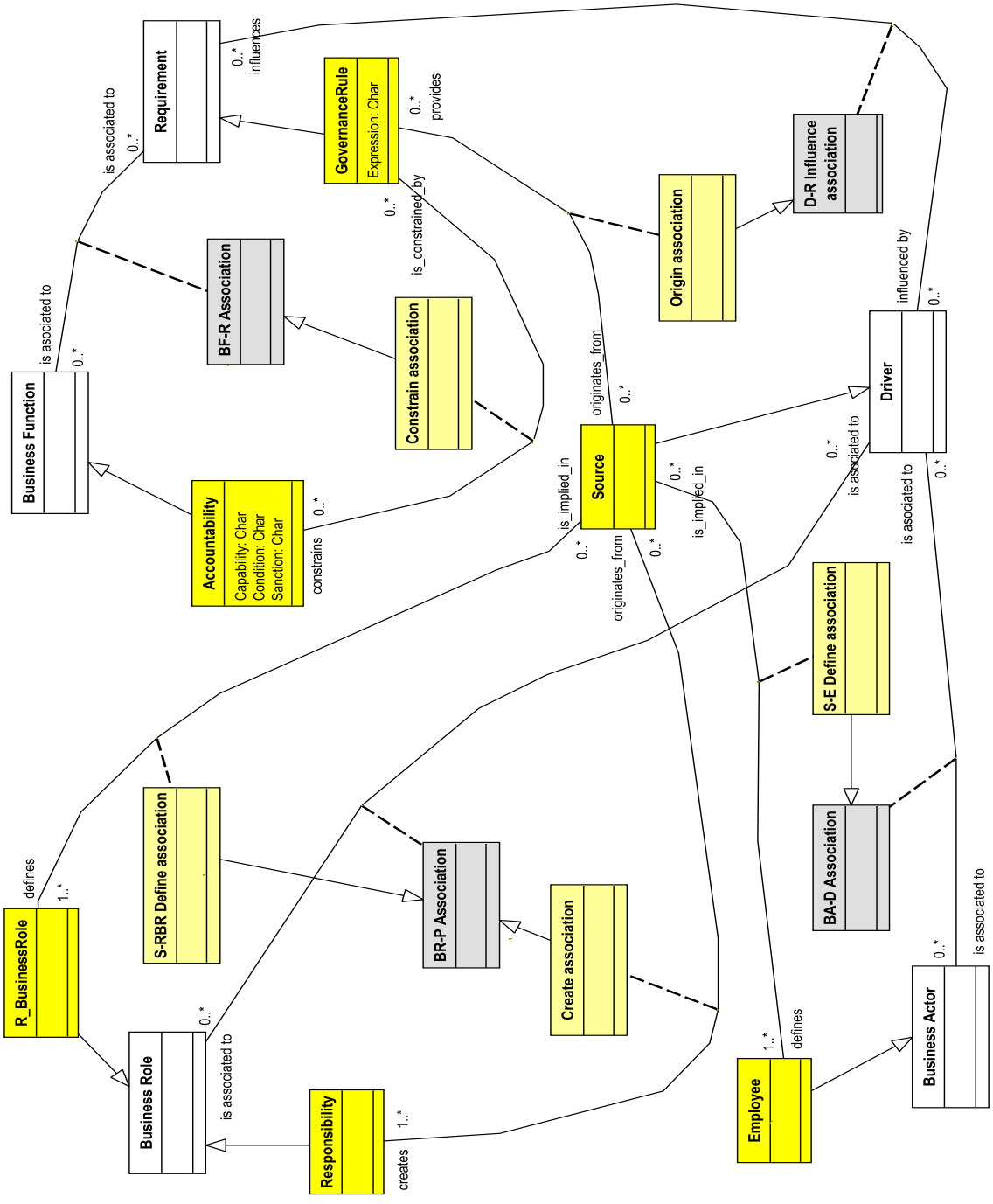


Figure 5.12: Source and Governance rule conceptual mapping

is a specialisation so that the `S-BRB Define association` and the `Create association` are specialisations of the `BR-D Association` class. This is represented, respectively, by ≪`S-BRB Define association`≫ and ≪`Create association`≫ are stereotypes of the `BR-D Association` class from ArchiMate.

Finally, in the Responsibility metamodel, the source is related to the employee with a `S-E Define association` class. In ArchiMate, the driver is related to the business actor with the `BA-D Association` class. We observe a mapping 1:1 between the both association classes and therefore, we apply the stereotype extension mechanism. The association between both classes is a specialisation so that the `S-E Define association` is a specialisation of the `BA-D Association` class. This is represented by ≪`S-E Define association`≫ as a stereotype of the `BA-D Association` class from ArchiMate.

5.4.7 Conceptual mapping summary

Table 5.5 provides a summary of the mappings realised between the elements (concepts and concepts' associations) from ArchiMate and ReMMo.

Responsibility element	**ArchiMate element**	**Mapping**	**Integration rule**	**Integrated element**
BusinessObject concept	Business object concept	1:1	Merge	Business Object
Task concept	Business process concept	1:1	Business process specialisation	≪Task≫
Task *Help* Task	Business process *aggregation* of Business process	1:1	BP-BP aggregation specialisation	≪Help≫
StructuralTask *Concern* BusinessTask	Business process *aggregation* of Business process	1:1	BP-BP aggregation specialisation	≪Concern≫
Task *Use* Business object	Business process *association* to Business object	1:1	BP-BO association specialisation	≪Use≫
R_BusinessRole concept	Business role concept	1:1	*ArchiMate* Business role specialisation	≪R_BusinessRole≫
Responsibility concept	Business role concept	1:1	Business role specialisation	≪Responsibility≫
Employee concept	Business actor concept	1:1	Business actor specialisation	≪Employee≫
Accountability concept	Business function concept	1:1	Business function specialisation	≪Accountability≫

Responsibility element	**ArchiMate element**	**Mapping**	**Integration rule**	**Integrated element**
Responsibility *Aggregation* of Accountability	Business role *association* to Business function	1:1	Association specialisation	≪Aggregation≫
Responsibility *Concern* BusinessTask	Business role *association* to Business process	1:1	Merge	BR-BP Association
Employee *Play* R_BusinessRole	Business actor *assignment* to Business role	1:1	Merge	BR-BA Assign association
Employee *Check* Accountability	Business actor *association* to business function	1:1	BF-BA association specialisation	≪E-A Check≫
R_BusinessRole *Check* Accountability	Business role *association* to business function	1:1	BF-BR association specialisation	≪RBR-A Check≫
Task *Perform* by Accountability	Business process *aggregation* of Business function	1:1	BP-BF aggregation specialisation and addition of attribute	≪Perform≫, Relation type: to_do or to_achieve
Employee *Assign* to Responsibility	Business actor *association* to Business role	1:1	Merge	BR-BA Association
R_BusinessRole *Assign* to Responsibility	Business role *association* to Business role	1:1	Merge	BR-BR Association
RightToUse concept	*access* association	1:1	Access specialisation	≪RightToUse≫
Sanction concept	–	–	Addition of attribute	≪Accountability≫, Sanction: Sanction description
Condition concept	–	–	Addition of attribute	≪Accountability≫, Condition: Condition description
Capability concept	–	–	Addition of attribute	≪Accountability≫, Capability: Capability description
Source concept	Driver concept	1:1	Driver specialisation	≪Source≫

Responsibility element	**ArchiMate element**	**Mapping**	**Integration rule**	**Integrated element**
GovernanceRule concept	Requirement concept	1:1	Requirement specialisation	≪Governance Rule≫
GovernanceRule *Origin* Source	Driver *influence* Requirement	1:1	D-R influence specialisation	≪Origin≫
GovernanceRule *Constrain* Accountability	Requirement *association* to Business function	1:1	BF-R association specialisation	≪Constrain≫
Source *Define* R_BusinessRole	Driver *association* to Business role	1:1	BR-D association specialisation	≪S-BRB Define≫
Source *Define* Employee	Driver *association* to Business actor	1:1	BA-D association specialisation	≪S-E Define≫
Source *Create* Responsibility	Driver *association* to Business function	1:1	BR-D association specialisation	≪Create≫

Table 5.5: Responsibility elements with ArchiMate elements mapping summary

Figure 5.13 represents the mappings between the concepts and associations between concepts from the Responsibility metamodel and ArchiMate metamodel.

5.4.8 Illustration

Figure 5.14 presents the Responsibility metamodel, instantiated according to the healthcare domain case study presented in Figure 4.12, modelled following the ArchiMate metamodel integrated with the Responsibility metamodel. To keep the figure straightforward, it does not include the `Sanction` neither the relations ≪needs/helps_for≫ and ≪concerns/is_concerned_by≫ between tasks, ≪is_related_to/is_related_to≫ between the sanction and the accountability and ≪checks/towards≫ between the business role and the accountability.

Overall, the Figure 5.14 is more easily readable and comprehensible than Figure 4.12, and is conforming to the ArchiMate formalism. The left side of the figure shows the source, the governance rule, the business roles and the employees. The middle of the figure shows the tasks and accountabilities. The right side of the figure shows the capabilities and the business objects.

5. CONCEPTUAL MAPPING AND INTEGRATION BETWEEN THE RESPONSIBILITY METAMODEL AND ARCHIMATE BUSINESS LAYER

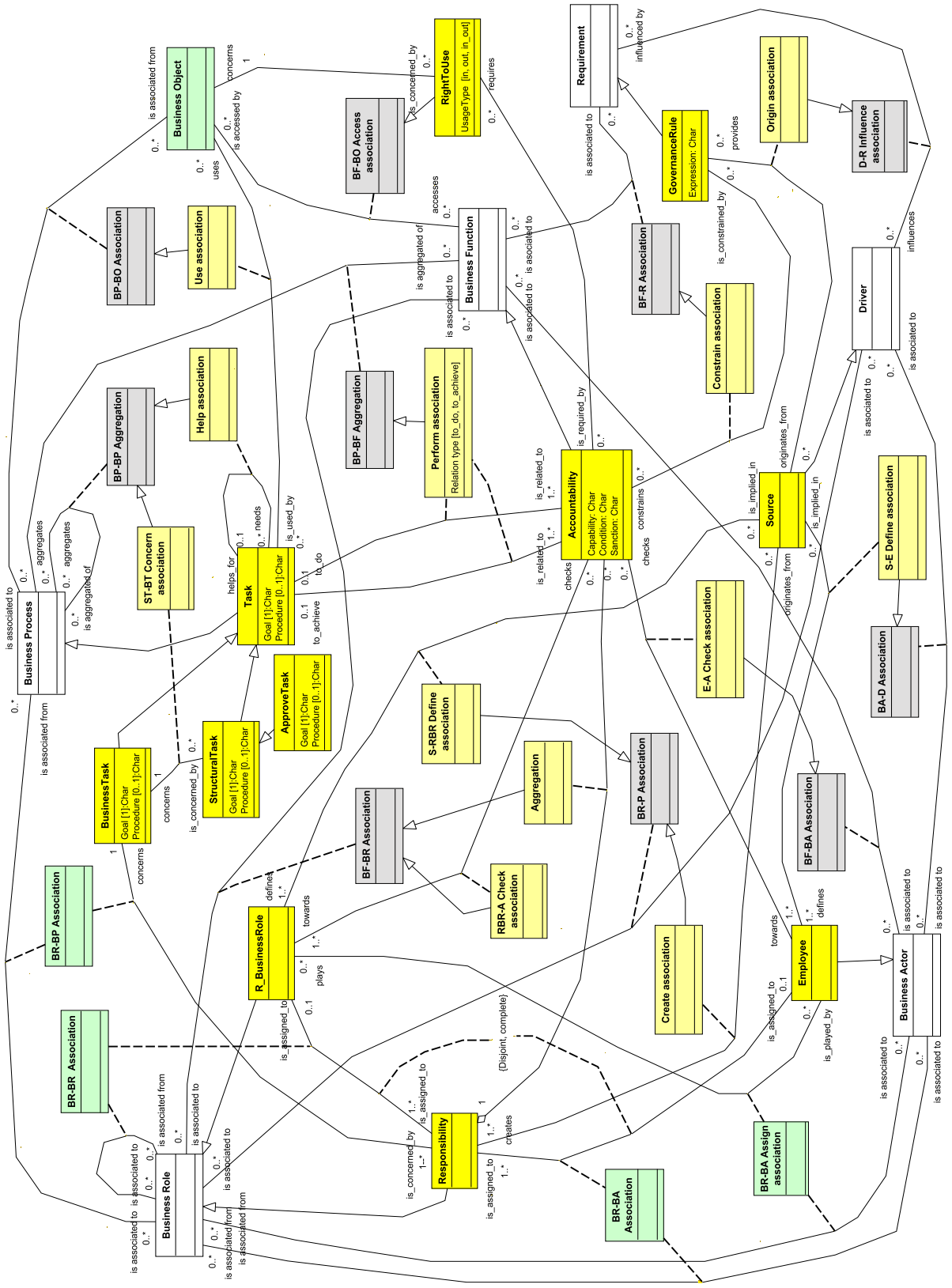


Figure 5.13: ArchiMate with the Responsibility metamodel conceptual mapping

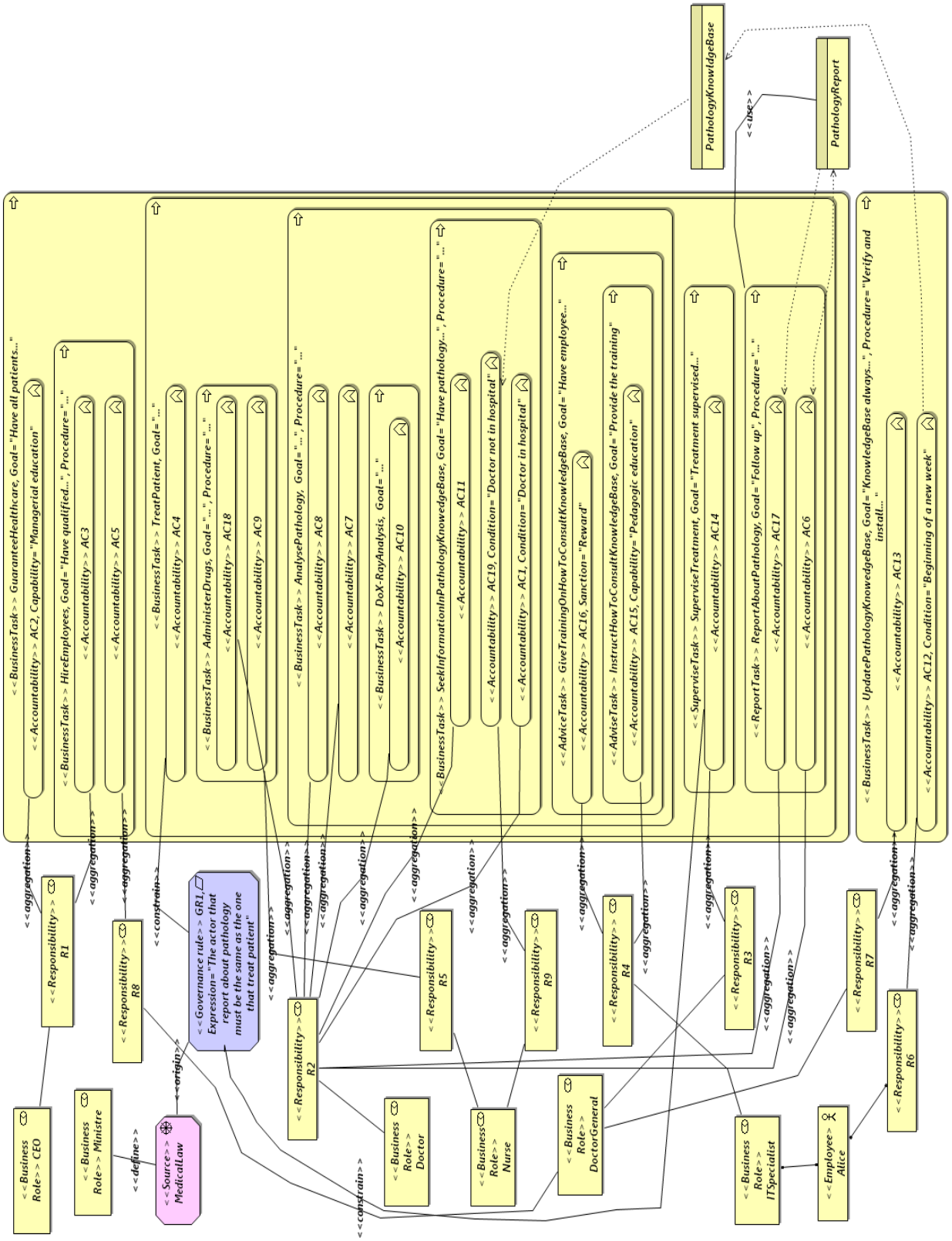


Figure 5.14: Responsibility metamodel instantiated in the healthcare domain following the conceptual mapping between ArchiMate and the Responsibility model

5.5 Case study at the Centre Hospitalier de Luxembourg. First part.

The second part of the chapter is dedicated to the first part of the case study at the Centre Hospitalier de Luxembourg. In this case study, we have instantiated ArchiMate and its Responsibility extension with **the activity to provision the access rights to the patient's record according to the employees' role and the hospital's specific access control model**. The objective of the case study was twofold: (1) to evaluate the expressiveness of the Responsibility metamodel and (2) to illustrate how the integration of ReMMo and ArchiMate may be instantiated to a real case.

The case study has been realised during one year, from January 2011 to January 2012. During this year, about eight meetings where organised. During the meetings, Patrick Recht, Responsable Support Application – Service Informatique[1], has provided a set of scenarii for accessing the patient records according to the different roles and based on the confidentiality data model related to these patient records. Scenarii is the expression used in the hospital and corresponds to an ordinate set of detailed steps. The analyse of the latter has been explained in Section 5.5.2 and has permitted to engineer the responsibilities that have been presented in Section 5.5.3.

At the end of this first part of the case study, a meeting of two hours was organised to evaluate the Responsibility metamodel and the integration of this Responsibility metamodel and RBAC in ArchiMate. During this meeting, ReMMo has been presented, as well as the integration of it and of RBAC in ArchiMate. Afterwards, we have reviewed the responsibilities that have been engineered from the scenarii, we reviewed how the scenarii are represented using the responsibilities.

This case study is presented in four steps. Firstly, we analyse the context and the scenarii for the access rights management in the hospital in Section 5.5.2. Then, in the Section 5.5.3, based on data collected in Section 5.5.2, we have modelled the different tasks, and responsibilities for these tasks, for each roles. Finally, in the Section 5.5.4, we have modelled the scenarii according to ArchiMate and its Responsibilibly extension. Finally, in Section 5.5.5, we have evaluated the results with Patrick Recht.

5.5.1 The Centre Hospitalier de Luxembourg

The hospital is a public institution focused on the care of severe pathologies, the medical and surgical emergencies, and the palliative care. The hospital also has an academic research character. In 2010, the hospital admitted 427,903 patients for consultations and outpatient visits, 25,532 inpatients, 33,277 adults emergency patients and 31,857 and paediatric emergency patients. At staff level, the hospital employed 2,046 employees including 152 physicians and specialised employees, 55 medical specialists who are liberal licensed, 53 cooperating physicians and 48 physicians in a specialisation process. The nursing staff included 1,336 employees and the administrative staff included 510 employees.

[1]Manager for the Application Support, IT department

The activity of the hospital is very unique due to the fact that the hospital is accessible 24 hours a day, 7 days a week and that refusing to assist a patient is not allowed, either during the day, night or weekend. During the night, the staff is reduced and a patient who arrives in the emergency room is received by the doctors on call. On weekends, the organisation of the emergencies in the Grand Duchy of Luxembourg is spread over its three regions and one hospital is allocated to each region. This means that each hospital receives, every two weeks, all the emergencies in its region and, therefore, its volume of activity is much more important.

In hospitals, the access to information may, in some cases, be critical for the life of the patient and refusing to provide the medical staff with the required access rights is, therefore, extremely complicated. On the other hand, these rights must be provided taking into account the respect of the patient's privacy. In this context, it is obvious that providing the medical staff with the appropriate accesses to the patient's records, at the right moment and without superfluous administrative duties, is crucial to the lives of these patients. The management of access rights related to this information is clearly a crucial activity.

To face these requirements, the hospital has developed its own access rights management based on the rule that the medical staff who accesses the patient record must be associated with this patient, and if this is not the case, he must motivate the intervention that justifies the access. Although, afterwards, this motivation is potentially subject to control, this is not always evident. For example, a lung specialist may require and thus request access to the psychiatric record of a patient since some lung problems may be triggered by neurological problems and as a consequence require neurological care. If an emergency doctor makes a diagnosis at entry, he may request a patient's record when a surgical operation is necessary and in this case, he motivates its access by "*EMERGENCY*".

5.5.2 Context of the hospital related to the access rights management

The access rights management of the hospital is based on a data model for the patient's records and on a set of scenarii to access these patient's records based on the existing roles in the hospital.

5.5.2.1 The data model for the patients' records of the hospital

The medical staff of the hospital consists of: (1) the doctors and the medicals secretaries who form the medical speciality and who are managed by the doctor general, (2), the nurses who form the care unit and who are managed by a chief of unit, and (3) the care healthcare specialists who form a healthcare service and who are managed by a chief of service.

The data model for the patients' records management of the hospital is structured in four levels (Figure 5.15) such that the higher the level is, the more sensitive, and thus, confidential the information is:

Level 1. This level includes data which is transversal for the patient record. It includes the lab results, the blood analysis, the patient historical data, and so forth. This information is available to the medical staff associated to the treatment of the patient. Level 1 is divided in two sub–levels: the information that is sensitive (level 1a) and the information that is highly

sensitive (level 1b), e.g., when a patient is HIV+. The sensitive information (level 1a)(e.g., patient historical data) is accessed after having been duly justified as for the level 2 (see below). The highly sensitive information of level 1b is the information which is judged more sensitive, by the medical staff, than the sensitive information of level 1a. Employees who are not from the healthcare domain have access to level 1a but not to level 1b, even with a justification. This is, for instance, the case of the receptionists.

Level 2. This level includes more sensitive information than the information on level 1 but less sensitive than that of level 3. This information is accessible to all but the access must be justified. In practice, a list of justifications is available for the medical staff and additional free text is also be available to complete this list. The justifications recorded by the medical staff are analysed afterwards based on different criteria like the frequency of the occurrences of a type of justifications (a doctor always provides the same justification) or the frequency of the accesses to a type of document. E.g. to detect if a doctor always uses the first justification of the list. The justifications are additionally analysed based on their semantic by a controller with medical knowledge.

Although the justification is subsequently subject to possible verification, this verification is not always easy. For example, a pulmonologist may require access to the file of a patient treated in the neurological speciality given that certain lung diseases occur due to neurological problems. Another observation is that the medical records must be immediately available. For example, when an emergency doctor makes the diagnosis of a patient in critical conditions, the motivation is simply "EMERGENCY".

Level 3. This level includes information more sensitive than the information on level 2 but less sensitive than that of level 4. The information at this level is available for all doctors and medical secretaries from the same speciality in which the patient is treated and to medical staff from other medical specialities, services or units, after:

1. an explicit association to the patient by the doctor in charge of this patient (note that all doctors from a speciality are associated to the patient in this speciality)

2. an explicit motivated request and the agreement of the doctor general of the service or its equivalent in a care unit or in a healthcare service.

Level 4. This level includes information more sensitive than the information of level 3 and includes data that is only accessible by the doctor associated to the patient. The doctor is, as a consequence, advised not to store information crucial for the life of the patient at this level. Only level 4 matches the medical confidentiality requirements.

Levels 2, 3 and 4 correspond to a vertical classification of the information based on their sensibility and according to a unique medical speciality, a care unit or a healthcare service such as the dietician or the psychologist.

In addition to these four levels, the doctor has the possibility to associate personal notes, concerning the patient, which are not necessarily linked to medical duties. With regards to the information stored in one of the four levels above, the personal notes cannot change from one level to another. If a doctor is deceased or leaves the hospital, the information is recovered by an alternative doctor although the personal notes are erased.

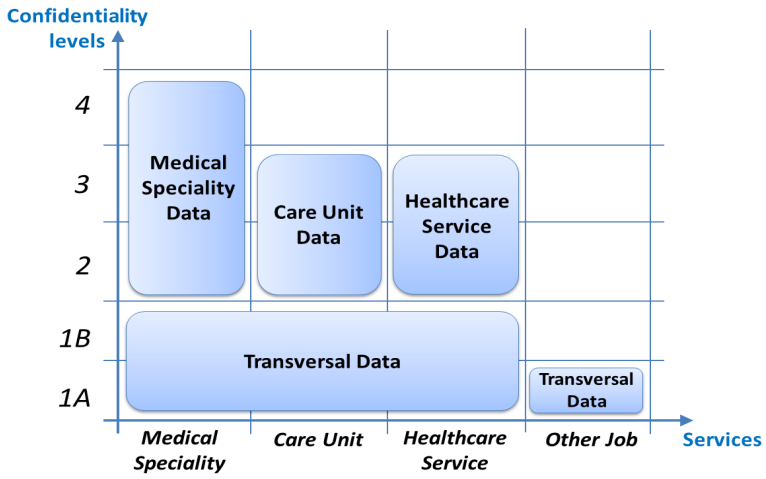


Figure 5.15: Data model of the hospital, function of the types of data, the services, and the confidentiality levels

5.5.2.2 Analysis of the scenario related to the doctors

There exists two types of role for the doctors: the doctor and the doctor general.

The doctor from speciality A who is associated to the treatment of a patient in speciality A has access to all levels of the patient's record, without justification for the access to the level 2 and without access request and agreement for the access to the level 3, but with a justification for level 1b. In practice, this association is obtained implicitly after an external admission of the patient in the doctor's speciality or after his transfer in this speciality. This transfer corresponds to an internal admission and is effective as soon as an order entry is delivered by a doctor's or a nurse associated to this patient. This *order entry* corresponds to a job request that includes the unit and the medical speciality in which the patient is affected.

When the doctor from the speciality A asks for advices from a colleague from speciality B regarding a precise medical case, he must associate the advisor doctor to the treatment of the patient. Afterwards, the advisor doctor may access level 2 of the data model in the speciality A with a justification and to the level 3 in this speciality after having made a request to the doctor general of the speciality A. In the mean time, the advisor doctor may access all levels of the patient's record in its speciality and may create personal notes.

One doctor in the same speciality as the doctor associated to the treatment of a patient is also automatically associated to this patient. In this case, the doctor accesses the information at all layers of the data model except 1b and may create personal notes at level 4 in this speciality. The data access at level 2 does not need to be justified and no request nor agreement is necessary to access data at level 3. However, the other doctor may not access the personal notes of another doctor, even from the same speciality.

In case of a *clinical order*[1], only the doctor from one speciality is associated to the treatment

[1]The clinical order is used, in the hospital, to order services for a treatment

of a patient. However, if this doctor creates data related to his speciality, all other doctors from this same speciality may access it. This clinical order may be delivered by a medical secretary.

In some cases, one doctor from speciality B needs to access the patient's record related to speciality A. In this case, the doctor from speciality B must ask a doctor associated to the patient in speciality A to be associated to the treatment of this patient. Moreover, if the doctor from speciality B needs to have access to the patient's record at level 3 in speciality A, he must make an explicit request to the doctor general of the speciality A. This situation often happens when a patient is transferred from speciality A to speciality B. In this case, all the doctors from speciality B have access to the patient's record related to speciality B at level 2 without justification and at level 3 without any explicit request. But, if one doctor from the speciality B also needs access to the patient's record at level 2 and 3 from speciality A, he must justify it or make an explicit request to the doctor who is doctor general of the speciality A.

The doctor who needs access to a patient's record but who is not associated to the treatment of this patient nor is in the same speciality as the doctor associated to the treatment of the patient may only access the patient's record at level 1 and 2 and all accesses are traced. This is the case for the emergency doctors or for the doctors on call who need to treat patients who they do not know. The model developed in the hospital supposes that these doctors never need to know information from level 3. In each case, they may create personal notes at level 3.

In the data model, the medical staff is not supposed to insert fundamental information at level 3. In practice, in the hospital, it has been observed that 90 percent of the information is created in level 1.

The doctor general of a speciality has to manage this speciality and in, thereby, responsible of the treatment provided to all the patients of this speciality.

5.5.2.3 Analysis of the scenario related to the medical secretaries

Three types of roles exist for the medical secretaries: the medical secretary, the typing pool medical secretary, and the multi–purpose medical secretary. We only consider the first two types in the case study.

The activities of the medical secretary include the assistance in making appointments, in managing the documentation, and in preparing the medical files and the patient's records. They also check and verify the data, and perform the patient transfer from one speciality to another.

Like the doctor, the medical secretary must be associated to the treatment of a patient and must be associated to a medical speciality to get access to the patient's records in this speciality. This association to the patient is automatically realised when the medical secretary makes an appointment for consultation or an hospital admission.

Due to their activities, a medical secretary needs to access confidential data. E.g., on the admission form, the secretary has to check boxes that correspond to a type of medical act. During the consultancy, the doctor also checks boxes which, afterwards, are recorded by the medical secretary in SAP. Therefore, they have access to the information of the three lower levels (except

level 4) of the data model. If a medical secretary wishes to get access to level 1b, she has to motivate her request. If the medical secretary fills out a document template, this document is automatically recorded and associated to a confidentiality level fixed by the doctor. Most of the time, only level 1a and 2 are concerned. The medical secretary is not allowed to change the level of a file. Only the doctors are allowed to perform this task.

With regards to the assignment to a speciality, a doctor will always remain in the same speciality (due to its education in a medical speciality), but a medical secretary can be transferred from one speciality to another. This means that a medical secretary is associated to a speciality when a request is formulated by a service and when, as a result, the speciality is attributed by the administrator. This is the case, for instance, when a medical secretary from speciality A needs information from speciality B to prepare a report, he/she must motivate the need to access the data at level 2 and must formally ask access to data at level 3 to the doctor general from this speciality B.

In some cases, a medical secretary will definitely change the service (speciality). In this case, all of the rights are modified by the information system.

The typing pool medical secretary is a particular type of medical secretary who, unlike the other medical secretaries, does not change working environment. They always remain physically in their office and are always busy recording data from many specialities. As a consequence, they are often requested to change speciality and as a result, they do not have to motivate the change of speciality. However, this change is traced by the system.

5.5.2.4 Analysis of the scenario related to the nurses and healthcare specialists

Two groups exist: the nurses and the healthcare specialists. The first group concerns the care provided within a precise care unit (generally located at the same geographic area) and the second group concerns the care provided through many units and concerns, e.g., the pain group, the dieticians, the physiotherapists or the psychologists. The nurses are associated to a patient based on the care unit and the healthcare specialists are associated to a patient based on the healthcare speciality that they provide.

Nurses and healthcare specialists also need to be associated to a patient to access the patient's record. This association is automatic when a patient is associated to a unit or to a healthcare speciality.

In general, a patient is associated to one or more medical speciality(ies), to a care unit and sometimes to one or more a care speciality(ies). The patients from the same medical speciality are often gathered in the same care unit but it is not always possible, and, as illustrated in Figure 5.16, some patients treated in the same medical speciality may sometimes be spread in more than one care unit.

The activities of the nurses include providing healthcare to the patients such as administering drugs, washing the patients, or if needed feeding them. The activities of the care specialists vary depending on the speciality. We consider that the activity of the healthcare specialist is to provide special care.

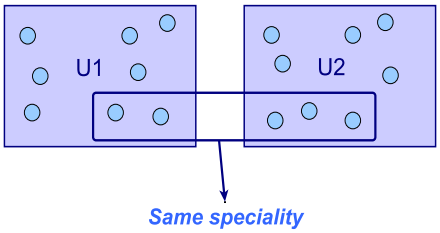


Figure 5.16: Links between care units U1 and U2, and a speciality

The information generated by the nurses and the healthcare specialists is different from the information generated by doctors but it is also spread over the three law levels (1, 2 and 3) of the data model, with the level 1 subdivided into a and b depending on the information sensitivity.

A nurse can create information in level 1, 2 and 3 related to the unit he/she is assigned to and all the nurses from this unit can read this information. When a patient is transferred from care unit A to care unit B, all the nurses from B receive the access to all the information from B although all the nurses from A receive access to the information concerning the care unit B only after justification for the information recorded in level 2, and after a motivated request for the information in level 3.

To move from one care unit to another (for instance because of its competencies), a nurse has to provide a motivation which could be, e.g., *Madam X calls me for help for one hour.*

5.5.2.5 Analysis of the scenario related to the quality analysts or the statisticians

The quality analysts access level 1a with traceability to the patient's information history and to the statistic environment (ETHNOS). The statisticians may not access confidential information and they are provided with information considering the anonymity of the patients. They may not access level 1b.

The statisticians only receive access to ETHNOS and not to the hospital enterprise resource planning software (namely SAP) so it is impossible to gather information like the patients names and private information.

5.5.3 Responsibilities modelling based on the scenarii

In Section 5.5.2, we have, for each roles, collected a set of scenarii which include a number of tasks for which the roles are responsible. In the sequel of this section, the second step consist in modelling these responsibilities using ArchiMate integrated with the Responsibility metamodel.

The entire set of tasks and responsibilities for the scenarii is the following:

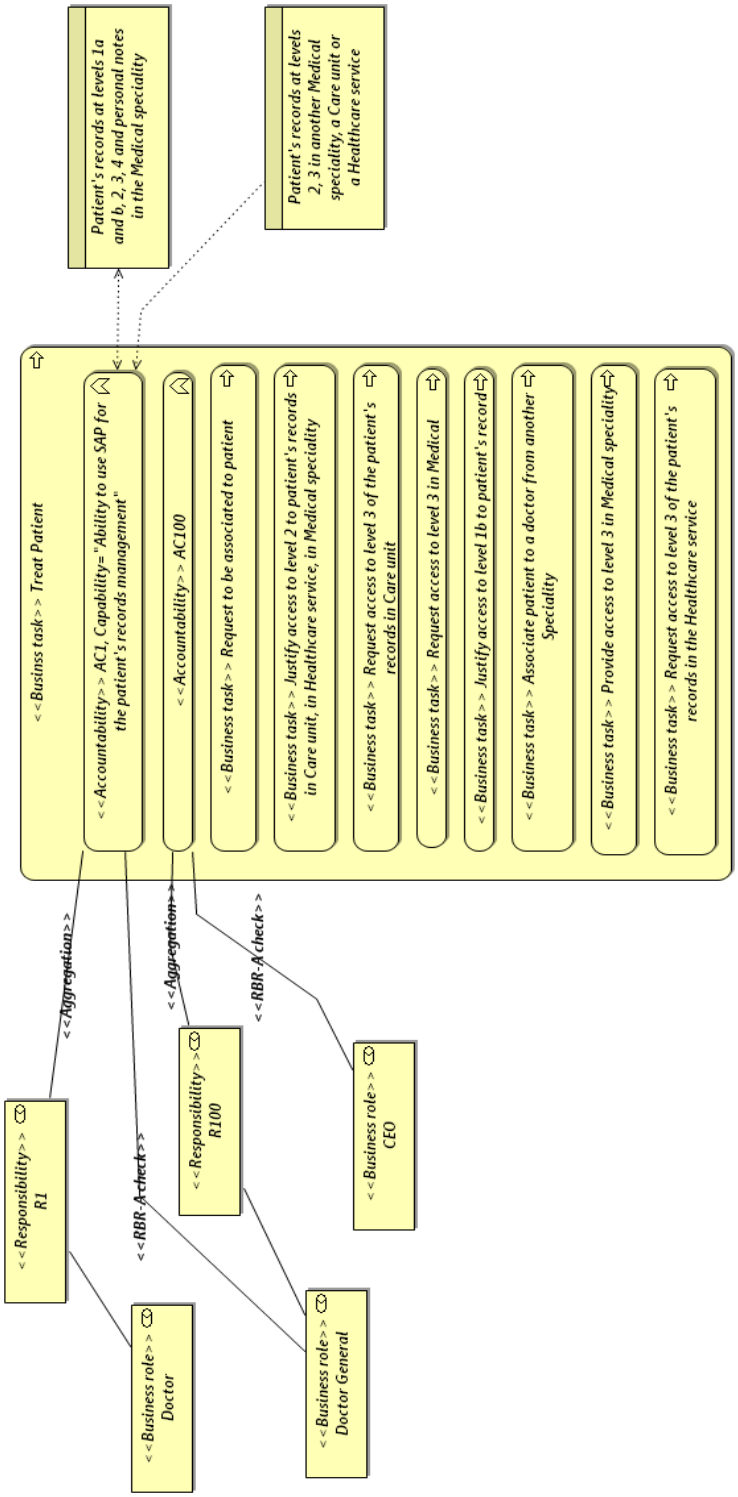


Figure 5.17: Responsibilities *R1* and *R100*

5.5.3.1 Tasks form the doctors scenario

From the doctors scenario, ten business tasks have been identified and modelled. Hereafter, in Figure 5.17, the ≪Business Task≫ Treat Patient is explained, for illustration. The other tasks are represented in Appendix A.1 to A.10.

- Treat patient (Figure 5.17)

  The task to treat a patient is modelled by ≪Business Task≫ Treat Patient which is a stereotype of a business process from ArchiMate and which is concerned by the ≪Responsibility≫ R1 and R100 which are stereotypes of the business role. ≪Responsibility≫ R1 is assigned to the ≪Business Role≫ Doctor and aggregates the ≪Accountability≫ AC1 which is a stereotype of a business function and which has as attribute the type of accountability to_do ≪Business Task≫ Treat Patient. This ≪Accountability≫ AC1 requires access in read/write mode the data object Patient's record at level 1a and b, 2, 3, 4 and personal notes in the doctor medical speciality and requires to access in read mode the data object patient's record at level 2 and 3 in another medical speciality. ≪Accountability≫ AC1 also requires the ≪Capability≫ to Use SAP for the patient's record management, which is a stereotype of business function. ≪Responsibility≫ R100 is assigned to the ≪Business Role≫ Doctor General and is included with the ≪Accountability≫ AC100 which has as attribute the type of accountability to_achieve ≪Business Task≫ Treat Patient.

- Request to be associated to a patient (make an explicit request) (Figure A.2)
- Associate a patient in the speciality to a doctor from another speciality (Figure A.3)
- Request access to level 3 of the patient's records in a medical speciality (Figure A.4)
- Provide access to level 3 of the patient's records in a medical speciality (Figure A.5)
- Justify access to level 1b of the patient's records (Figure A.6)
- Justify access to level 2 of the patient's records in a care unit, in a healthcare service, or in a medical speciality (Figure A.7)
- Audit of the justification to access level 1b and level 2 of the patient's records in a care unit, in a healthcare service, or in a medical speciality (Figure A.8)
- Request access to level 3 of the patient's records in a care unit (Figure A.9)
- Request access to level 3 of the patient's records in a healthcare service (Figure A.10)

5.5.3.2 Tasks from the medical secretaries scenario

From the medical secretaries' scenario, six business tasks have been identified and modelled. These models are in Appendix B.1 to B.6.

- Assist a doctor from a speciality (Figures B.1 and B.2)
- Transfer a patient from one speciality to another (Figures B.3 and B.4)

- Request to be punctually assigned to a medical speciality (Figure B.5)
- Motivate the punctual assignment to a medical speciality (Figure B.6)

5.5.3.3 Tasks from the nurses scenario

From the nurses' scenario, four business tasks have been identified and modelled. These models are in Appendix C.1 to C.3.

- Provide care to the patient (Figure C.1)
- Change unit (Figure C.2)
- Motivate the unit change (Figure C.4)
- Provide access to level 3 of the patient's records in a care unit (Figure C.3)

5.5.3.4 Tasks from the healthcare specialists scenario

From the healthcare specialists' scenario, two business tasks have been identified and modelled. These models are in Appendix C.5 and C.6.

- Provide special care to the patient (Figure C.5)
- Provide access to level 3 of the patient's records in a healthcare service (Figure C.6)

5.5.3.5 Tasks from the quality analysts and statisticians scenario

From the quality analyst and statistician's scenario, two business tasks have been identified and modelled. These models are in Appendix D.1 and D.2.

- Manage quality (Figure D.1)
- Make statistics (Figure D.2)

5.5.4 Modelling of the scenarii using ArchiMate extension with the Responsibility metamodel

Step three aims at modelling the scenarii collected in the analysis of the context performed in the first step using the tasks and the responsibilities which have been elaborated in the second step. To illustrate this modelling, we have selected two scenarii: the first one is very simple and concerns the nurse that changes unit, and the second one is more complete and concerns the doctor who treats a patient and one doctor that advises the doctor that treats the patient.

5.5.4.1 Change of unit

The first scenario concerns a nurse who performs a change of unit (Figure 5.18). This scenario is only composed of the ≪Business Process≫ Change of unit. The ≪Business role≫ Nurse is associated to the ≪Responsibility≫ R31 which aggregates the ≪Accountability≫ A31 which has the attribute to_do the ≪Business Task≫ Change of Unit. This accountability is modelled in Figure C.2. To realise the latter, the nurse requires the ≪Capability≫ to Use SAP for creating a change of unit request and the right to write a Change request business object.

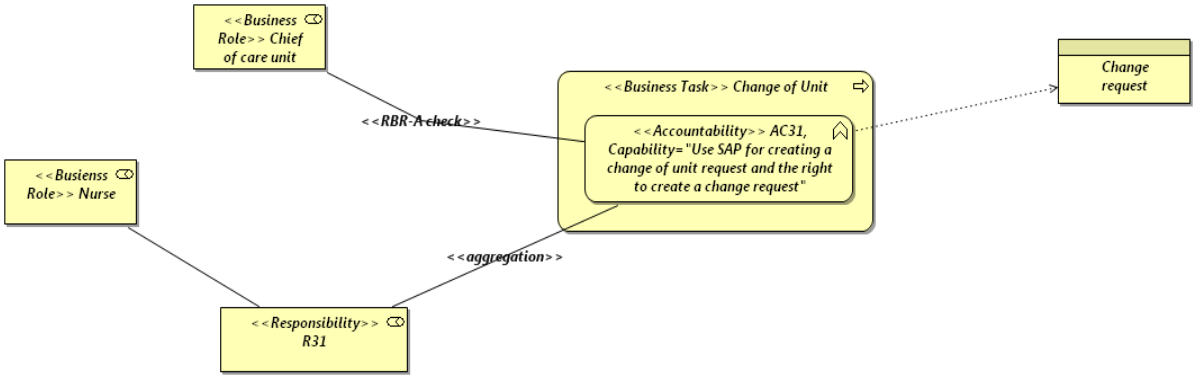


Figure 5.18: Scenario *Change of Unit*

5.5.4.2 Treat a patient

The second scenario concerns a doctor from the surgery discipline who's responsibility is represented by the ≪Responsibility≫ R1a and which aggregates the ≪Accountability≫ AC1 which has the attribute to_do the ≪Business Task≫ Treat Patient in its speciality (Figure 5.19). To perform this business task, the ≪Business Role≫ Doctor requires the access in read/write mode to the Patient's records at levels 1a and b, 2, 3, 4 and personal notes in the Surgery which is a data object. Moreover, for the treatment of this patient, he requires the access in read mode to the level 3 of the patient's record in Pulmonology to know about his patients' lung conditions. Therefore, two additional ≪Business Task≫ are modelled in the scenario: (1) Ask access to level 3 of the patient's record in Pulmonology. This ≪Business Task≫ aggregates the ≪Accountability≫ AC4 defined in Figure A.4. The realisation of this ≪Business Task≫ helps for the ≪Business Task≫ Treat Patient and (2) Provide access to level 3 of the patient's record in Pulmonology. This ≪Business Task≫ aggregates the ≪Accountability≫ AC5b defined in Figure A.5. The decision for providing access or not composes the ≪Responsibility≫ R5b and is assigned to the ≪Business Role≫ Doctor General of Pulmonology Service.

To treat a patient, the doctor also needs advice from the pulmonologist. Therefore, we create the ≪Responsibility≫ R1b assigned to the ≪Business Role≫ Doctor of Pulmonology and which aggregates the ≪Accountability≫ AC1b which has the attribute to_do the ≪Structural Task≫ Give advices. To give advice, this doctor is also accountable (≪Accountability≫ AC4b)

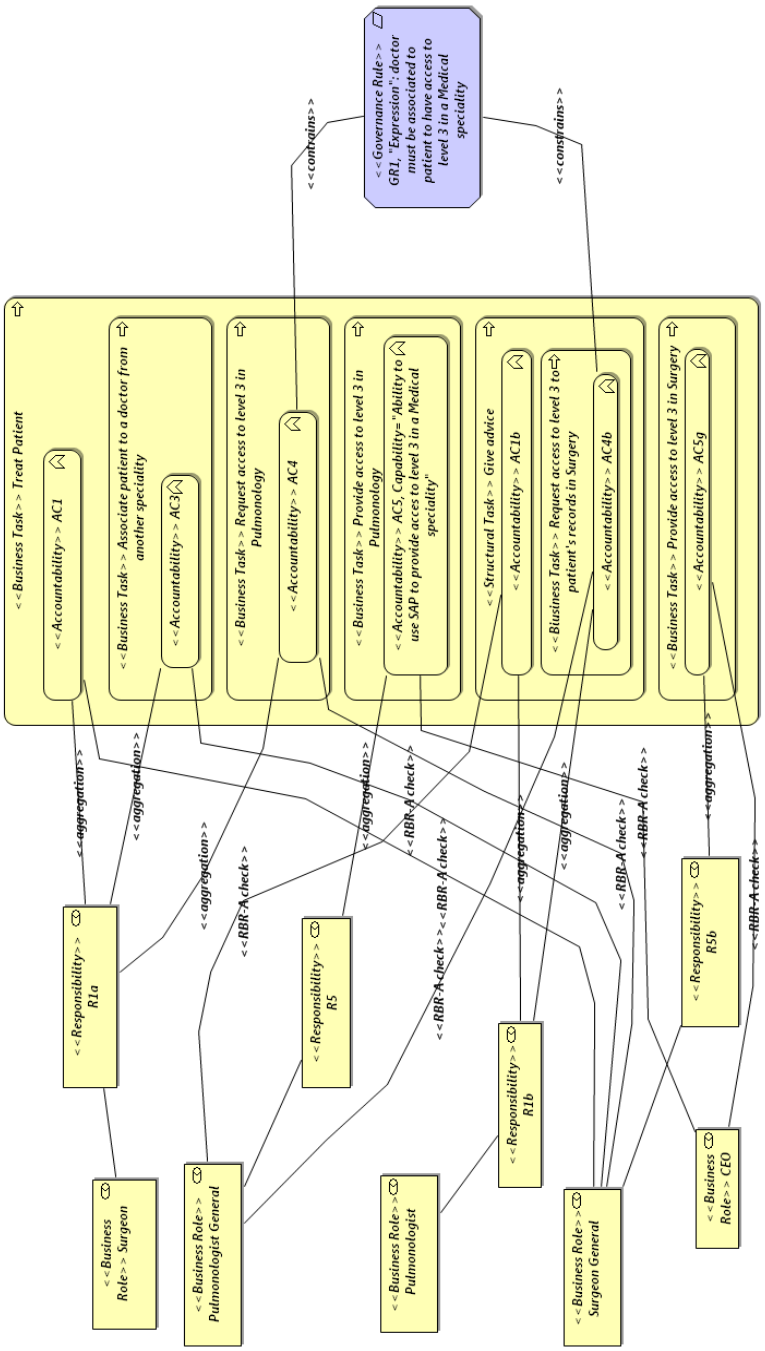


Figure 5.19: Scenario *Treat a Patient in Surgery*

to request access to level 3. Previously, the ≪`Business Role`≫ `Surgeon` which is assigned to the ≪`Responsibility`≫ `R1a` which aggregates the ≪`Accountability`≫ `AC1` and which has the attribute `to_do` the ≪`Business Task`≫ `Treat Patient` must have associated the patient to the pulmonologist (≪`Accountability`≫ `AC3`) and the surgeon general must provide access to level 3.

The list of associations of doctors to patients is a business object, and the governance rule for a doctor to access level 3 is to be associated to the patient. This is modelled using a stereotype of the ArchiMate motivation extension of requirement: ≪`Governance Rule`≫.

5.5.5 Evaluation of the first part of the case study

Patrick Recht was interviewed on the issue of the first part of the case study. He estimates that he has a very good knowledge of the enterprise architecture field and a good knowledge of information security. According to him, considering the responsibilities of the employees: (1) could enhance the performance of the hospital, (2) is appropriate to perform business IT/alignment and (3) is feasible and realistic to deduce the access rights to be provided to the employees.

Regarding ReMMo, he estimates that the relevance of the concepts that compose the metamodels is good and that considering the employees accountabilities in this metamodel is justified. Patrick, additionally, proposed the following enhancements to the metamodel:

- **Provide type of Employees**.
  To better understand the metamodel, it could be interesting to explain that the concept of employee includes the *Manager*. Explaining this could facilitate the understanding that the accountability is due from one employee to a manager, which is a type of employee.
- **Highlighting the importance of the Means**.
  Patrick Recht considers that the *Means* is an important concept which may be required by an accountability to_do or to_achieve a task. This *means* is, for instance: the budget available for the employee accountable for the task, the time necessary to perform it, the possibility to be helped by a team of co–workers, and so forth. Adding this concept of means in the Responsibility metamodel as a type of capability sounds justified indeed.

5.6 Conclusions

In Chapter 4, we have defined a Responsibility metamodel which we integrate in this chapter with the business layer of the ArchiMate enterprise architecture. To realise the integration, we have analysed the correspondences and resolved the semantic and structural heterogeneities between the concepts (or associations between concepts) from the Responsibility metamodel and from the core ArchiMate business layer and motivation model extension.

In the second part of the chapter, we have illustrated the ArchiMate extension with ReMMo by a real case study in a hospital. The case study has been elaborated in three steps. Firstly, we have collected a set of scenarii for different business roles: doctor, medical secretary, nurse and healthcare specialist, and quality analyst and statisticians. In parallel, we have also collected the data model of the hospital. Secondly, we have modelled the responsibilities that we

recovered in the scenarii and we have associated the business roles analysed in the case study with these responsibilities. Thirdly, we have described two modelled scenarii in order to highlight how they can be represented using ArchiMate extended with the Responsibility metamodel.

This case study has allowed us to illustrate that it is possible to use ArchiMate extended with the Responsibility metamodel to model the solution designed by the hospital to access the patient' records. In Chapter 6, we will explain how the integration of the two metamodels will serve the provisioning of the access rights to be provided to the employees at the application layer of ArchiMate. Thereby, in Chapter 6, we will highlight how the responsibility is usable as a pivot to enhance the alignment between both layers.

Publication related to this chapter:

- C. Feltus, E. Dubois, E. Proper, I. Band, M. Petit, Enhancing the ArchiMate® Standard with a Responsibility Modeling Language for Access Rights Management, in *Proceedings of the 5th International Conference on Security of Information and Networks (SIN)*, Jaipur, Rajastan, India. 2012. ACM.

Chapter 6

Alignment between the access rights management and the Responsibility management

6.1 Introduction

In Chapter 5, we have mapped and integrated the Responsibility metamodel with the business layer of ArchiMate. This integration has allowed the formalisation of the responsibilities of the business actors and has been evaluated using a case study at the Centre Hopitalier de Luxembourg. In this case study, we have evaluated the expressiveness of ReMMo and the usability of its integration with ArchiMate to formulate the responsibilities of the employees of the hospital regarding a set of scenarii related to the management of the access rights. In Chapter 5, the provisioning of the access rights according to these responsibilities has not been tackled yet. This is the objective of this chapter.

In the field of the access rights, the state of the art presented in Chapter 2 has highlighted that many access control models exist to improve the management of these rights. The most used is RBAC (Ferraiolo et al. (2001)), which has already been included in many existing and widespread applications. Given this statement, RBAC is the access control model that we are going to consider in the following.

The management of the access rights, based on RBAC and using the enterprise architecture approach, presents many potential advantages such as the possibility to align the access rights to be provided to the users, at the application layer, with the rights they really require at the business layer, to perform business processes. However, despite these interests, we note that the concepts from the business layer of ArchiMate are roughly aligned with the concepts from the RBAC model, exploited at the application layer. This is mainly due to the lack of appropriate concepts, at the business layer, to precisely define and motivate the assignments of permissions to users. Given this weakness, in this chapter, we propose a method based on ArchiMate extended with ReMMo to engineer and optimise the assignment of permissions to employees according to their responsibilities. Therefore, in Section 6.2, we remind how RBAC currently exists in, and may be modelled by, ArchiMate. To that end, we present the previous work realised by

Band (2011) related to the definition of a *RBAC reference model* at the application layer. Then, in Section 6.3, we analyse how the definition of the employee's responsibilities at the business layer could enhance the instantiation of RBAC at the application layer. Therefore, we align RBAC and the Responsibility metamodel, and we analyse which concepts from the Responsibility metamodel allow generating concepts from the RBAC model. Subsequently, based on this alignment, we propose an *Access rights management reference model* in Section 6.4. The latter includes, amongst others, five processes which contribute to populate the *RBAC reference model* proposed by Band.

In Section 6.5, with the second part of the case study in the Centre Hospitalier de Luxembourg, we evaluate that we may better engineer the access rights required by the employee considering their responsibilities. Therefore, along the case study, we firstly express the current situation in the hospital, secondly, we engineer, based on the elaboration of the business roles' responsibilities, what the required access rights should be, and thirdly we compare both situations and draw conclusions.

6.2 ArchiMate and the access rights management

Before realising the alignment between RBAC and the Responsibility metamodel, this section firstly recalls how the RBAC model exists through the enterprise architecture layer of ArchiMate, in Section 6.2.1. Secondly, it presents Band's *RBAC reference model*, at the application layer, in Section 6.2.2. This review aims at supporting the definition of the integrated metamodel for the access rights management.

6.2.1 RBAC model through ArchiMate layers

RBAC has been introduced in Section 2.2.3. As a reminder, RBAC is a high level model which has for objective simplifying the management of permissions to users provisioning. This is especially necessary in multinational companies where the amount of employees often goes in thousands. RBAC models the permissions which are assigned to the users according to the roles their are assigned to. As a result, RBAC can be defined by the three main following components: the user, the role, and the permission (Figure 2.3). According to the RBAC specifications, the concept of permission represents the realisation of a set of operations on objects. RBAC provides access decisions based on two associations: the association of users to roles based on the function which the users realise, and the association of permissions to roles. This means that with RBAC it is easy to change the assignment of employees to roles without changing permissions associated to the role.

At the business layer of ArchiMate, enterprises tend to be organised based on business roles that are assigned, on the first hand, to business actors and, on the second hand, to business processes. Moreover, a business process requires to access business objects. This association between the components of the business layer is similar to the RBAC model. This means that the business layer of the enterprise architecture is modelled according to the RBAC model to provide the business actors with accesses to the business objects. This statement is commonly agreed upon by the enterprise architecture practitioners. For instance, Gaaloul and Proper (2013) have recently proposed a solution for the management of organisational resources according

to ArchiMate. Therefore they have reasoned about task–based resources and have proposed a conceptual model supporting access control at the business layer. At the application layer, RBAC has also been embedded in many operating systems and applications as well, Cenys et al. (2009). This usage of the RBAC model at the application layer has also been corroborated by Band (2011) who, as explained in Section 6.2.2, has proposed a reference model to formally represent RBAC at the ArchiMate application layer. This *RBAC reference model* is explained in the next section.

6.2.2 Band's RBAC representation and management at the application layer of ArchiMate

For the representation and the management of RBAC based access rights management solutions, at the application layer of ArchiMate, Band (2011) proposed a *RBAC reference model* modelled by means of the existing core ArchiMate concepts. These concepts, which are exploited for this representation of RBAC at the application layer, are:

- The concept of **Data object** which represents *a passive element suitable for automated processing*,
- The concept of **Application function** which accesses the data object and represents *a behaviour element that groups automated behaviour which can be performed by an application component.*

Theses concepts are illustrated in Figure 6.1.

Figure 6.1: Data object and application function concepts from the application layer of ArchiMate, **Adapted from:** ArchiMate® 2.0 specifications (The Open Group (2012))

To represent RBAC at the application layer, Band has created a set of data objects which realises the concepts or associations between concepts from the business layer. This set of data objects that we consider in the sequel of this chapter and their meaning is summarised in Table 6.1. It is worth noting that only to the core RBAC model (or RBAC0, as explained in Section 2.2.3) is addressed in this thesis. The hierarchical and constraint RBACs are not analysed and may be addressed in future works.

Data Object	Meaning
`Users`	This data object realises implicitly the business actor from the business layer. This business actor is represented, at the application layer, by e.g., an ID.

Data Object	Meaning
`RBAC Roles`	This data object realises implicitly the business role from the business layer. The concept of role at the application layer corresponds to the role that is exploited by applications that use RBAC (Cenys et al. (2009)). For the sake of clarity, this `RBAC Role` is the name given to the `Role` data object from Band (2011)
`Permissions`	This data object realises implicitly an authorization to access a set of business objects.
`Users-RBAC Roles Assignments`	List of users assigned to a list of RBAC roles
`Permissions-RBAC Roles Assignments`	List of permissions assigned to a list of RBAC roles

Table 6.1: Data objects meaning

To perform the RBAC administration, Band relies on this set of data objects and defines a set of application functions for the assignment of users to roles and the review of this assignment, for the assignment of permissions to roles and the review of this assignment, for the management of the roles hierarchy, for the management of the separation of duties constraint, for the management of the sessions and the performance of access check. Band explains all of these functions in Band (2011). In the frame of this chapter, only two of these application functions which compose the `RBAC Administration` are exploited: `Assign Permissions to RBAC Roles` and `Assign Users to RBAC Roles`. These application functions are summarised in Table 6.2. We do not consider the other application functions, nor the data objects that they use.

Application Function	Meaning
`RBAC Administration`	Is the main application function which is composed of, amongst others, the functions: `Assign Permissions to RBAC Roles` and `Assign Users to RBAC Roles`
`Assign Permissions to RBAC Roles`	Instantiate the Permissions-RBAC Role Assignments data object considering the Permissions and the RBAC Roles data object
`Assign Users to RBAC Roles`	Instantiate the Users-RBAC Roles Assignments data object considering the Users, the RBAC Roles and the Session data object

Table 6.2: Application functions meaning

Practically, the administration of the access rights is performed by instantiating the data objects defined in the above Table 6.1 using the `RBAC Administration` main application function (Figure 6.2) and by performing access checks, according to the information represented by the data objects, using the `RBAC System Support` main function. Concerning the `RBAC administration`, this main function requests, on the one hand, to perform the assignment of `Users` to `RBAC Roles`, therefore the `Assign Users to RBAC Roles` application function reads the `Users` and the `RBAC Roles` data objects and writes the `Users-RBAC Roles Assignments` data object, and on the other hand, to execute the assignment of permissions to RBAC roles,

therefore the `Assign Permissions to RBAC Roles` function reads the `Permissions` and the `RBAC Roles` data objects and writes the `Permissions-RBAC Roles Assignments` data object. For his part, the `RBAC Support System` allows checking that an access may be granted to the users by reading the `Active Role Set` and the `Permissions to RBAC Roles Assignments`, and by comparing this with the access requested.

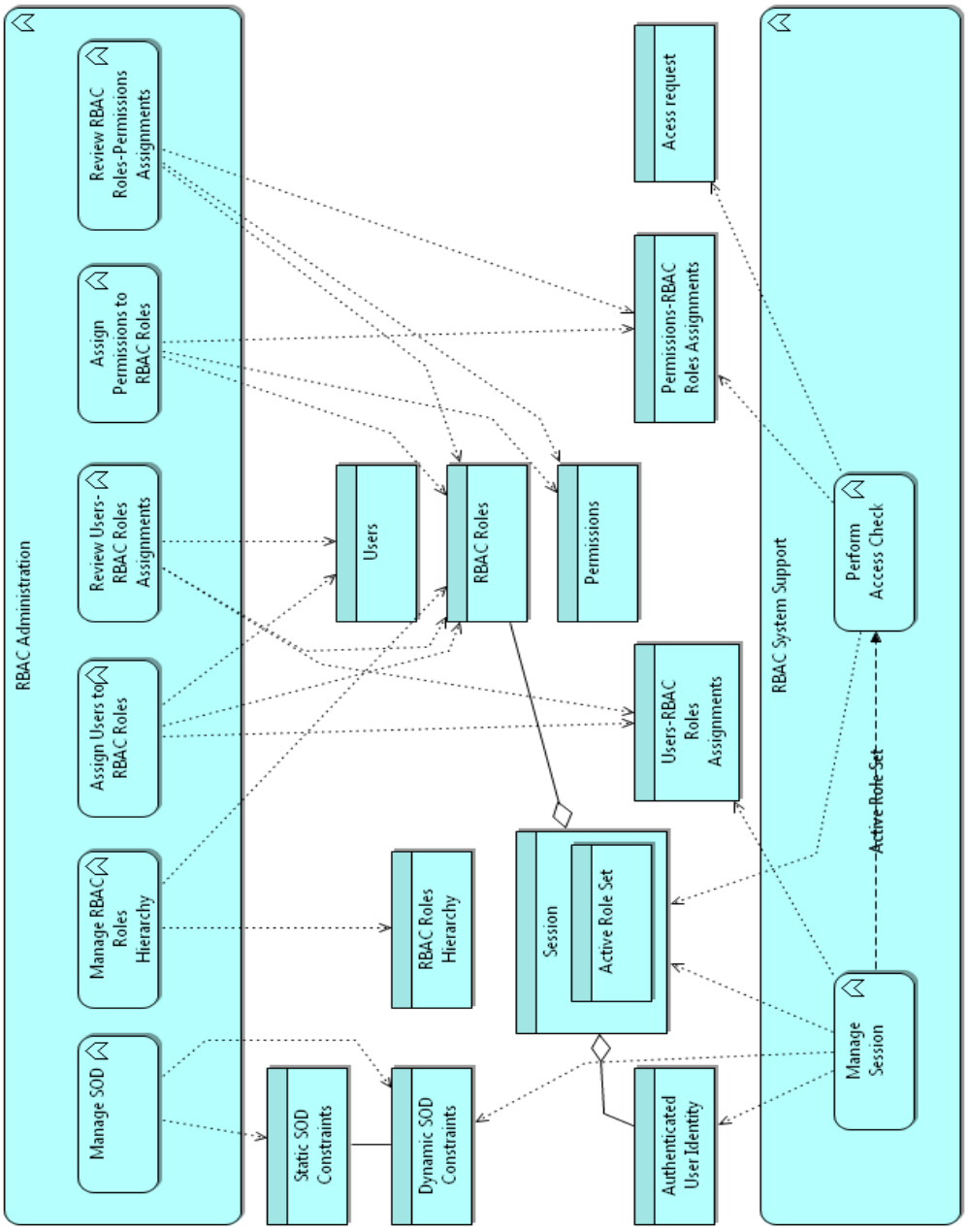


Figure 6.2: Band's *RBAC reference model*, representation of the RBAC concepts at the application layer of ArchiMate, **Adapted from:** Band (2011)

6.3 Alignment between RBAC and the Responsibility metamodel

This section aims at aligning ReMMo and RBAC with the objective not to elaborate an integrated metamodel but to figure out to what extend the elaboration of the responsibilities contributes to instantiate RBAC. The alignment is based on RBAC modelled in UML from Shin and Ahn (2000), Ray et al. (2004), Kim et al. (2004) and Anderson (2004). In Figure 6.3, the concepts from RBAC are represented in dark orange and the relations between concepts in light orange. The concepts from the Responsibility metamodel are in dark yellow and the relations between concepts in light yellow. To perform the alignment between concepts from RBAC and from the Responsibility metamodel, we exploit the *trace to association*. As defined by Object Management Group (OMG) (2011), the *trace to association* specifies *the trace relationship between model elements or sets of model elements that represent the same concept in different models. Traces are mainly used for tracking requirements and changes across models. Since model changes can occur in both directions, the directionality of the dependency can often be ignored. The mapping specifies the relationship between the two, but it is rarely computable and is usually informal.*

The following *trace to* relationships between concepts, and relations between concepts, are realised:

- The employee from the Responsibility metamodel is defined as a human entity that may or may not play one or more business roles. Depending on the business role played, the employee may require permissions on the information system. In RBAC, the user mainly represents a human. There exists a *trace to association* between the `User` class from RBAC and the `Employee` class from ReMMo. This is represented in Figure 6.3.

- The business role from the Responsibility metamodel may represent a set of employees who share common characteristics and are assigned to responsibilities. This business role may or may not require permissions on the information system. In RBAC, the role means a *job function with some associated semantics regarding the responsibilities conferred to the users assigned to it.* There exists a *trace to association* between the concept of RBAC Role from RBAC, which we represent by the `RBAC role` class in the following of this chapter, and the `BusinessRole` class from the Responsibility metamodel.

- In ReMMo, the employee is associated to the business role through the *Play association*. In the RBAC model, the user is associated to the RBAC role. There exists a *trace to association* between the `Play association` and `Users-RBAC Roles Assign association` classes. However, a RBAC role, generated from a business role is assigned to a user which is generated from an employee only if this business role is played by this employee. Therefore we introduce the constraint A.I which is:

  Constraint A.I: *A `RBAC Role` generated from a `BusinessRole` is assigned to a `User` generated from an `Employee` if this `BusinessRole` is played by this `Employee`*

- In the Responsibility metamodel, an employee may be directly assigned to a responsibility although in RBAC, a user may not directly be assigned to a permission. Practically, to

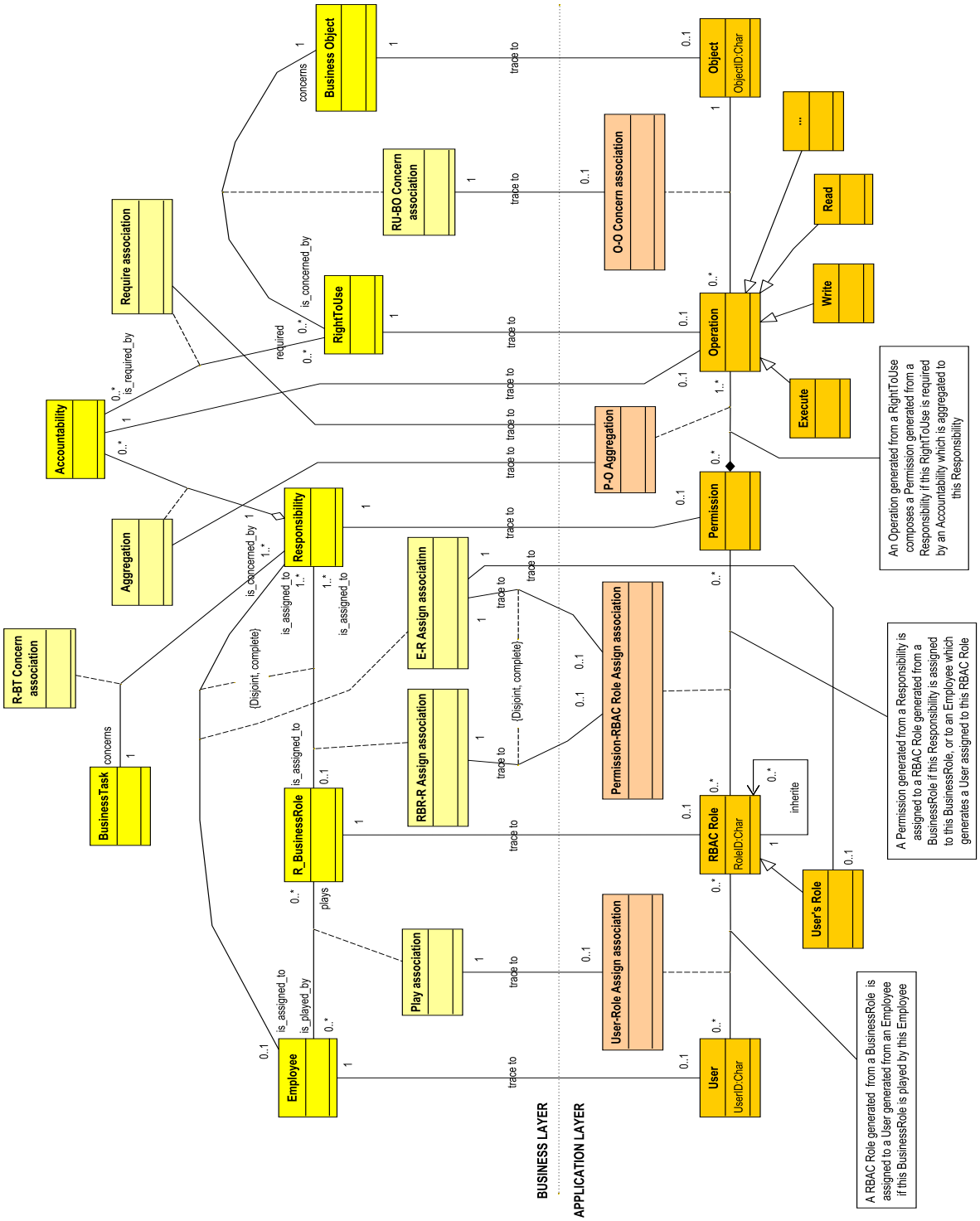


Figure 6.3: Alignment between RBAC and the Responsibility metamodel

realise this association with RBAC, we need to define a special RBAC role such as *only the user which corresponds to the employee directly assigned to the responsibility is assigned to this role*. This special RBAC role is named *User's Role*, is represented by the `User's role` class, in Figure 6.3, and corresponds to a specialisation of the `RBAC Role` class. This `User's Role` is generated to represent the `E-R Assign association` class. Therefore, we associate both concepts with a *trace to association*.

- In the Responsibility metamodel, the responsibility concerns a unique business task and aggregates a set of accountabilities which relates to this business task, to the task(s) needed by this business task, and to the structural task(s) concerned by this business task. These accountabilities require rights to use the business objects which are, themselves, used by the business task. In RBAC, according to Ray et al. (2004) a permission *determines which operations a user assigned to a role can perform on information resources*. Hence a permission encompasses a set of operations on business objects. Both the responsibility and the permission from RBAC have in common the gathering of a list of operations related to business objects. As a consequence, we observe a *trace to association* between the `Permission` class and the `Responsibility` class, as represented in Figure 6.3.

  Acknowledging this mapping, we observe that while there exists no justification nor guideline, in RBAC, for the gathering of a set of operations within a permission, the mapping of RBAC with ReMMo permits to justify that these operations are gathered according to the unique business task concerned by the responsibility.

- In the Responsibility metamodel, the responsibility is associated to the business role or to the employee through, respectively, the *RBR-R Assign association* or the *E-R Assign association*. In the RBAC model, the permission is associated to the RBAC role through the *Permission-RBAC Role Assign association*. We observe that this RBAC role is generated either by the RBR-R Assign association or by the E-R Assign association but not by both associations at the same time. Therefore, we create two *trace to associations*: (1) between the `Permission-RBAC Role Assign association` class from RBAC and the `RBR-R Assign association` class from the Responsibility metamodel and (2) between the `Permission-RBAC Role Assign association` class from RBAC and the `E-R Assign association` class from the Responsibility metamodel, and we express the constraint that these relations are *Disjoint* and *Complete*. Additionally, we observe that a permission, generated from a responsibility, is assigned to a RBAC role, generated from a business role, if this responsibility is assigned to this business role or to an employee which generates a user assigned to this RBAC role. This is expressed by the constraint A.II:

  Constraint A.II: *A `Permission` generated from a `Responsibility` is assigned to a `RBAC Role` generated from a `BusinessRole` if this `Responsibility` is assigned to this `BusinessRole`, or to an `Employee` which generates a `User` assigned to this `RBAC Role`*

- In the Responsibility metamodel, the right to use corresponds to an authorisation to perform an operation on a business object. In RBAC, a permission is defined as an approval of a mode of access to a resource. Hence, we consider that there exists a *trace to association* between the `RightToUse` class from ReMMo and the `Operation` class from RBAC.

- In RBAC, an object corresponds to an information object. In ReMMo, the business object is defined as a passive element which may be a document or an information. We consider that there exists a *trace to association* between the `Business Object` class from the Responsibility metamodel and the `Object` class from RBAC.

- Concerning the associations, we observe that the *BO-R Concern association* between the business object and the right to use from ReMMo generates the *O-O Concern association* between the operation and the object from RBAC. We represent this by a *trace to association* between the `BO-R Concern association` class from the Responsibility metamodel and the `O-O Concern association` class from RBAC.

- Finally, in the Responsibility metamodel, the responsibility aggregates accountabilities which require rights to use, although in RBAC, the permission aggregates operations. Practically, in the Responsibility metamodel, this is represented by the *Aggregation* and the *Require* associations, and in RBAC this is represented by the *P-O Aggregation*. We observe that the latter is generated by the *Aggregation* between the responsibility and the accountability, and by the *Require association* between the accountability and the right to use, from ReMMo. Therefore, we create two *trace to associations*. The first one is between the `P-O Aggregation` class and the `Aggregation` class and the second one is between the `P-O Aggregation` class and the `Require association` class. Additionally, we observe that an operation generated from a right to use composes a permission generated from a responsibility if this right to use is required by an accountability which is aggregated to this responsibility. This is expressed by the constraint A.III:

> **Constraint A.III**: *An `Operation` generated from a `RightToUse` composes a `Permission` generated from a `Responsibility` if this `RightToUse` is required by an `Accountability` which is aggregated to this `Responsibility`*

6.4 RBAC access rights management modelling in ArchiMate

Considering the integration of ReMMo with the business layer of ArchiMate performed in Chapter 5 and given the alignment of the Responsibility metamodel with RBAC performed in Section 6.3, this section proposes an *Access rights management reference model* suited to populate and to extend elements from the *RBAC reference model* proposed in Section 6.2.2 according to the responsibilities of the employees defined at the business layer of ArchiMate.

This *Access rights management reference model* is presented in Figure 6.4. The lower layer of it represents a fragment of the *RBAC reference model*, at the application layer, and the upper layer represents, at the business layer, the *Access rights management reference model* itself (this layer is named: *Access Rights Management*). The concepts from the *RBAC reference model* which are represented and which we want to instantiate are the users, the RBAC roles, the permissions, the users–RBAC roles assignments and the permissions–RBAC roles assignments. They were defined in Table 6.1. The *Access Rights Management* layer represents the access rights management processes which collects the information from the responsibilities of the employees, modelled with ArchiMate extended with ReMMo. This access rights management layer is composed of the *RBAC administrator* role, which is assigned to five business processes: *Populate the list of Users*, *Populate the list of RBAC Roles*, *Populate the list of Permissions*,

Populate the list of Users to RBAC Roles assignments and *Populate the list of RBAC Roles to Permissions assignments*. These fives processes are explained in the following:

- The first process, necessary to manage the access rights, populates the list of users. The user, as explained in Section 6.2.2, is a person from the subset of employees who requires to use the information system. We have analysed in Section 6.3 that the users are generated from the employees. As a result, this process aims at collecting, at the business layer of the enterprise, the list of employees who need to access the information system in order to perform the business process they are assigned to. Hence, in practice, this list of employees is collected from the responsibilities modelled with ArchiMate extended with the Responsibility metamodel. The result of the deployment of this process is a business object named *List of Users*. This is represented in Figure 6.4 by the `Populate the list of Users` class `write` the `List of Users` class. Afterwards, to be handled by the *RBAC reference model* at the application layer, this *List of User* business object is realised by the *Users* data object. This is represented by the `Users` class `realise` the `List of Users` class.

- The second process concerns the population of the list of RBAC roles. The RBAC role, as explained in Section 6.2.2, is a role from the subset of business roles which requires to use the information system. Given that the RBAC roles are generated from the business roles (Section 6.3), like for the population of the list of users, this process aims at collecting, at the business layer of the enterprise, the list of business roles which need to access the information system. Hence, such as for the employees, this list of business roles is collected from the responsibilities modelled with ArchiMate extended with the Responsibility metamodel. The result of the deployment of this process is a business object named *List of RBAC Roles*. This is represented in Figure 6.4 by the `Populate the list of RBAC Roles` class `write` the `List of RBAC Roles` class. Afterwards, to be handled by the *RBAC reference model* at the application layer, this `List of RBAC Roles` business object is realised by the *Roles* data object. This is represented by the `Roles` class `realise` the `List of RBAC Roles` class.

- The third process populates the list of permissions. Given the alignment of RBAC with ReMMo, we have analysed that the permissions are generated from the responsibilities assigned to the employees or to the business roles regarding a specific business processes. As a result, to populate the list of permissions, the RBAC administrator must collect, through the responsibilities modelled by the ArchiMate extended with the Responsibility metamodel, which rights to use are required to realise the responsibilities related to a specific business process. The result of the deployment of this process is a business object named *List of Permissions*. Practically, this is represented by the `Populate the list of Permissions` class `write` the `List of Permissions` class. Equally, to be handled by the *RBAC reference model*, this `List of Permissions` business object is realised by the *List of Permissions* data object. This is represented by the `Permissions` class `realise` the `List of Permissions` class.

- The fourth process populates the list of users assigned to RBAC roles. This process analyses the responsibilities modelled at the business layer of the enterprise in order to formalise the *List of Users to RBAC Roles assignments* which is a business object which represents which employee (who requires to use information at the application layer) plays which business role. At the business layer, this is modelled using the class `List of Users`

`to RBAC Roles assignments` which is related to the `Populate the list of Users to RBAC Roles assignments` class by means of a `write` relation. Additionally, to be handled by the *RBAC reference model*, at the application layer, this *List of users to RBAC roles assignments* business objects is realised by the *Users–RBAC Roles Assignments* data objects which is represented by the `Users-RBAC Roles Assignments` class `realise` the `Populate the list of Users to RBAC Roles assignments` class.

- The fifth and last process populates the list of permissions assigned to RBAC role. In the same way, this process analyses the responsibilities modelled business layer to formalise the *List of Permissions to RBAC Roles assignments* which is a business object which represents which permissions are required by the business role in order to use information existing at the application layer. Considering the alignment of RBAC with the Responsibility metamodel (Section 6.3), this list of permissions to RBAC roles assignments is populated from the association of responsibilities to business roles or to employees. At the business layer, this is modelled using the class `List of Permissions to RBAC Roles assignments`. This class is related to the `Populate the list of Permissions to RBAC Roles assignments` class by means of a `write` relation. Additionally, to be handled by the *RBAC reference model*, this *List of permissions to RBAC roles assignments* business objects is realised by the *Permissions–RBAC Roles Assignments* data objects. This is represented by the `Permissions-RBAC Roles Assignments` class `realise` the `Populate the list of Permissions to RBAC Roles assignments` class.

The modelling of the *Access rights management reference model*, at the business layer, illustrates how this access rights management may be decomposed into five processes, and how these processes write a set of dedicated business objects which are afterwards realised by data objects used to represent and relate the users, the RBAC roles and the permissions at the application layer. The population of these business objects is performed by the RBAC administrator which collects the information related to the employees, the business roles, and the responsibilities assigned to both, from the analysis of the business layer. Practically, this business layer is described and analysed through business processes documentation, job descriptions, or interviews of employees and managers.

The next section of this chapter illustrates the method for the access rights management based on the *Access rights management reference model* using the second part of the case study in the Centre Hospitalier de Luxembourg.

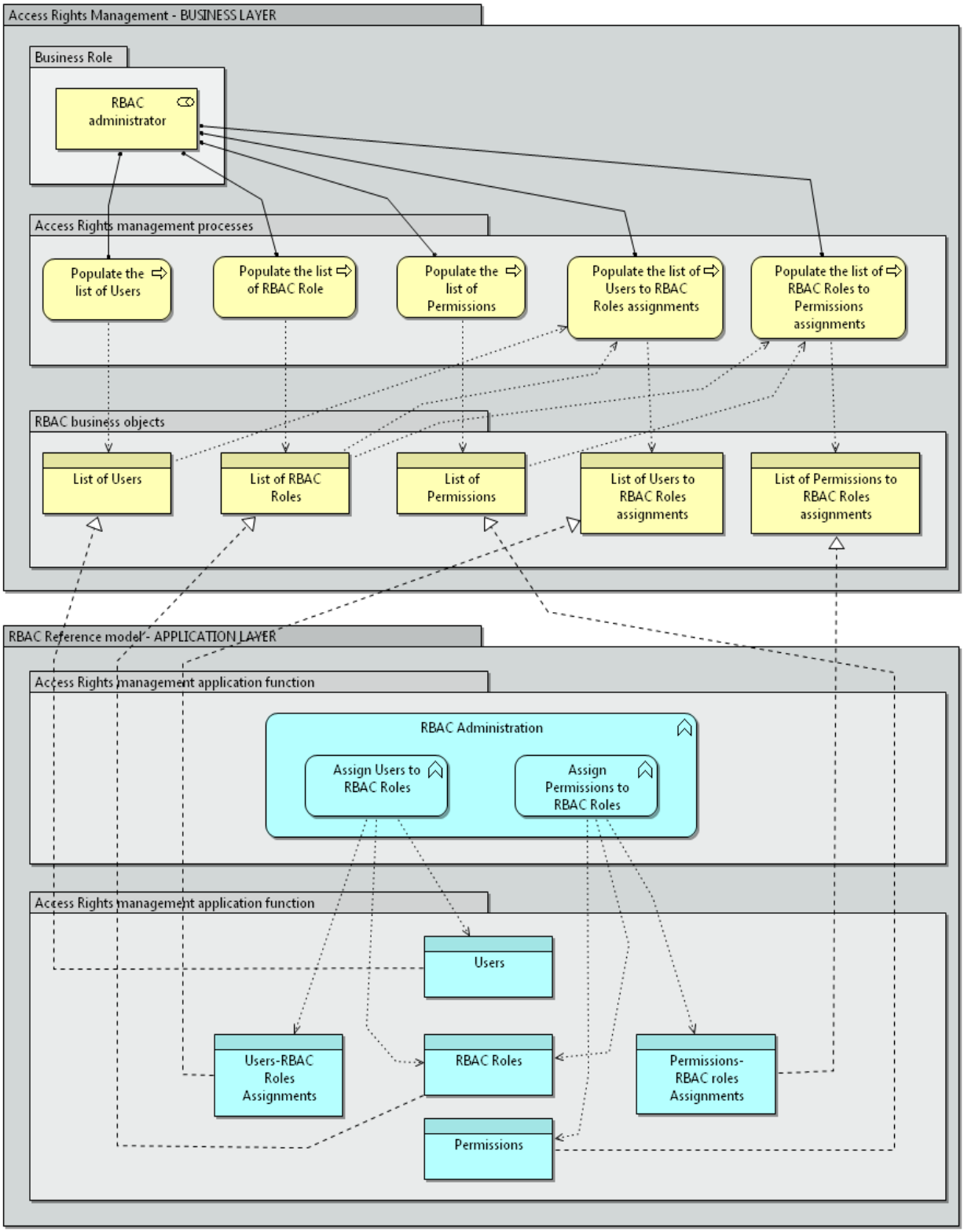


Figure 6.4: *Access rights management reference model*

6.5 Case study at the Centre Hospitalier de Luxembourg. Second part.

6.5.1 Scope and objectives of the case study

In the Centre Hospitalier de Luxembourg, the assignment of access rights is managed in different ways if it concerns the accesses to the patient's records or the accesses to other professional software. The first type of access rights management has been addressed in Section 5.5, has allowed to evaluate the expressiveness of the Responsibility metamodel, and has allowed to illustrate the integration of ReMMo with the business layer of ArchiMate to define responsibilities. The second type of access rights management is addressed, and will be explained, in the second part of this case study. The two following objectives are targeted: evaluate that the integrated ArchiMate with ReMMo, at the business layer, enhances the definition of the access rights required by the business role, and evaluate that the definition of the responsibilities at the business layer may be used to generate the RBAC roles and permissions, at the application layer. In practice, this is illustrated with the *Réceptionniste d'Accueil* role from the hospital, which we translate from French into English in *Receptionist role*.

This case study was realised during the months of February and March 2012. During this period, four meetings of two hours were organised with Frank Schmitz, Responsable du service accueil[1], and Laurent Wehr, Chargé de la gestion des compétences[2]. During these meetings, we have analysed the receptionist role, defined the responsibilities, and analysed the rights to use required for each of the accountabilities aggregated by these responsibilities. Therefore, Frank Schmitz and Laurent Wehr have provided the information necessary to understand the responsibilities of the employees working in the reception department and Marco Pappafava, Ingénieur Support Applications Service Informatique[3] has provided the list of existing RBAC roles and permissions.

The case study is structured as follows. In Section 6.5.2, we analyse the existing rights management activity in the hospital and we extract the list of permissions actually assigned to business roles. In Section 6.5.3, we deploy the method based on the *Access rights management reference model*, proposed in Section 6.4, and we defined new permissions to be assigned to the identical business roles. Finally, in Section 6.5.4, we compare the permissions provided to those really required. At the end of the second part of the case study, an evaluation meeting was also organised with Frank Schmitz and Laurent Wehr, and the result obtained during the case has been evaluated. During these meetings, the Responsibility metamodel was presented, the responsibility engineered have been evaluated, as well as the their mapping with the business roles from the receptionist. We have also presented the actual set of permissions provided to the employees and the rights they should really be assigned to according to their responsibilities.

[1] Reception department manager
[2] Manager of the competences management
[3] Applications Support Engineer IT Services

6.5.2 Existing access rights management in the hospital

In this section, we review and illustrate, for the reception roles, the existing activity of access rights management in the hospital to collect the permissions actually provided to the business roles. Therefore, we firstly explore the business layer to gather the list of existing business roles, in Section 6.5.2.1, than we explore the application layer to depict the existing RBAC roles, permissions, and assignments of those permissions to the RBAC roles, in Section 6.5.2.2. Finally, we collected the relations between the business roles and the RBAC roles, in Section 6.5.2.3. These relations allow, as a result, engineering which permissions are assigned to which business roles.

6.5.2.1 Existing business roles

At the business layer of the hospital, the employees are categorised based on their business roles. Some of these business roles have already been provided in the first part of the case study (Section 5.5). An organisation chart for the reception department splits the activities into eight business roles, which are:

- BR1: Receptionist at the *Clinique d'Eich* and at the *Hôpital Municipal*
- BR2: Receptionist at the paediatric clinic and at the maternity
- BR3: Phone reception
- BR4: Infodesk
- BR5: Human resources management
- BR6: Department management
- BR7: Room operator
- BR8: Outsourced guardian

6.5.2.2 Existing RBAC roles, permissions and assignments amongst both

At the application layer, the architecture of the information system of the hospital is composed of vertical software and transversal software (Figure 6.5).

1. Vertical software are the applications used by well defined and well specified healthcare businesses. Thesy are, for instance: the management of the laboratory, the endoscopy software, or the management of the polyclinic.

2. Transversal software are used by all healthcare businesses. They are, for instance: the dispatching of the laboratory's results or the medical imaging. The hospital ERP is the most important transversal software.

The hospital ERP is a business management software that offers the possibility to programme specific application functions by the owner of the application himself. Therefore, it has been decided by the hospital to use it to manage the access rights to all the other software. As a

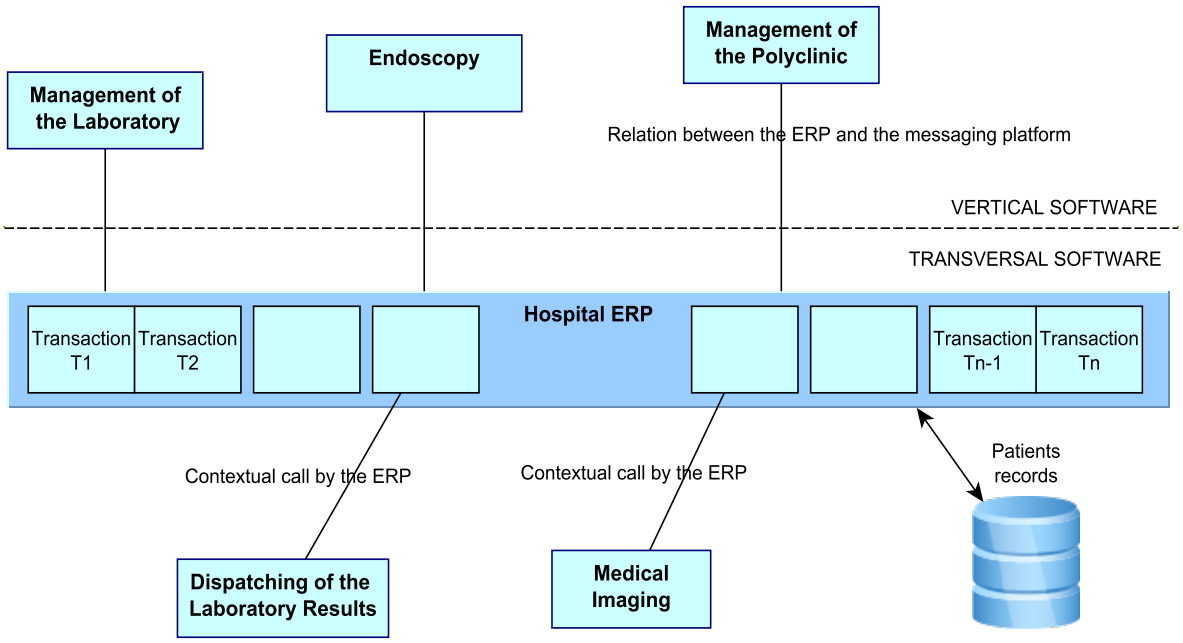


Figure 6.5: Software architecture of the hospital

consequence, there exist connections between the ERP and the vertical software, on the one hand, and connections between the ERP and the other transversal software, using contextual calls, on the other hand. Using this ERP, the access rights management is realised by means of *AuthorityObject* (e.g., in Figure 6.6). These *AuthorityObject* are composed of zone(s) from 1 to n based on which authority checks are performed. In practice, *AuthorityObject* correspond to ERP transactions (Figure 6.5) and for each of them, a set of authorisations is defined such as create, modify, delete, view historic, and so forth.

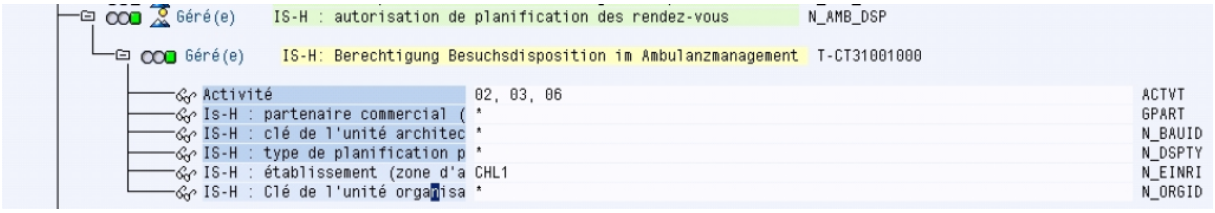


Figure 6.6: Example of interface to manage the *AuthorityObject*: N_AMB_DSP

The *AuthorityObject* illustrated in Figure 6.6 has for objective to formalise the authorisation N_AMB_DSP of the employees related to the operation *schedule appointment*. This authorisation concerns the predefined types of operations 02, 03 and 06 which corresponds, respectively, to modify, display or delete an appointment. The authorisation is specified using five zones which allow performing authorisation checks. These zones are:

- GPART: commercial partner. E.g., doctor, physiotherapist, and so forth.

- N_ BAUID: consultation room. E.g., a0.3, c3.1, and so forth.
- N_ DSPTY: duration of the consultation. E.g., 10, 15 or 20 minutes.
- N_ EINRI: concerned building. E.g., main hospital or care building somewhere in the city.
- N_ ORGID: organisational unit. E.g., cardiology, pulmonology, gastrology, and so forth.

To facilitate their management, *AuthorityObject* is assigned to *Functional Roles* like, for instance, the *Functional Role* to search for a patient in the database, create a patient entry, create a transaction, show a transaction, and so forth. Additionally, the concept of *Reference user* has been created to gather a set of *Functional Roles*. In practice, one user may be assigned to 1 or more *Reference user* or to one or more *Functional Role*.

As illustrated in Figure 6.7, the access rights model from the hospital, at the application layer, is similar to the RBAC model. At the concepts layer, we observe that the `Functional Role` and the `Reference User is_a RBAC Role` and that the `AuthorityObject is_a Permission`. Additionally, we note that there exists a role hierarchy between the `Functional Role` and the `Reference User` such as the second `inherits` the `AuthorityObject` assigned to the first. In the following of this case study, we use the RBAC vocabulary of RBAC role and permission when we refer, respectively, to the *Functional Role*, the *Reference User* and the *AuthorityObject*.

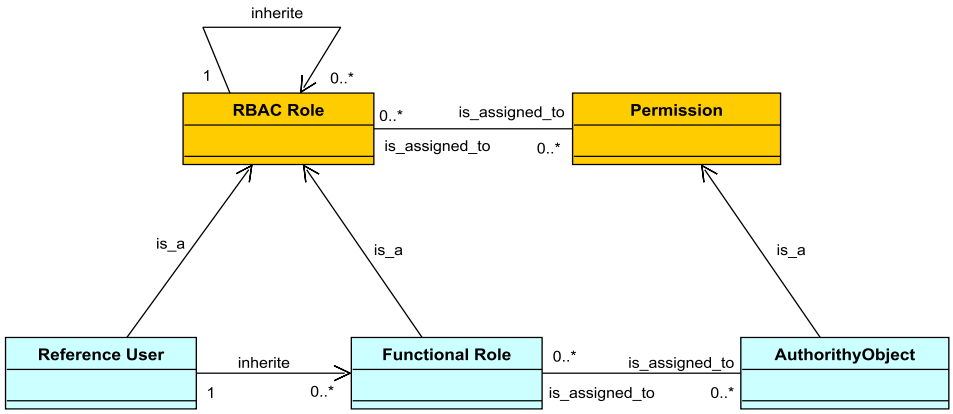


Figure 6.7: UML representation of the employee assignment to *Reference User* and *Functional Role*

In the IT department of the hospital, an authorisation profile document formalises the RBAC roles that may be assigned to the users from the reception. These RBAC roles are assigned to permissions, as expressed in Table 6.3.

RBAC roles	**Permissions**	
	Operations	**Objects**
RR1	Create, add, modify, display, delete	Basic patient's data
RR2	Create, add, modify, display, delete	Entry, transfer or leaving of patient's data
RR3	Create, add, modify, display, delete	Bed status file

RBAC roles	Permissions	
	Operations	**Objects**
RR4	Create, add, modify, display, delete	Medical delivery data
RR5	Create, add, modify, display	patient's invoices record

Table 6.3: List of RBAC roles to permission assignments

In practice, some specific permissions may directly be assigned to certain employees. For the clarity of the case study, we need to define a workaround to have a hospital's access rights model fully compliant with the RBAC model. Therefore we create six additional specific RBAC roles which we assigned to these additional specific permissions. These six new RBAC roles are assigned to permissions as expressed in Table 6.4.

RBAC roles	Permissions	
	Operations	**Objects**
RR6	Create, add, modify, display, delete	Data in the equipment ordering software
RR7	Display	Planning of doctors on duty file
RR8	Create, add, modify, display, delete	Data in the Excel file: *Timetable planning*
RR9	Create, add, modify, display	Data in the reporting software
RR10	Create, add, modify, display, delete	Data the room agenda in GroupWise multi–users software
RR11	Create, add, modify, display, delete	Data in the statistical software

Table 6.4: List of specific RBAC roles to permissions assignments

Finally, a twelve RBAC role named REFRECEPTION also exists in the hospital and inherits the permissions assigned to the RBAC roles RR1, RR2 and RR3.

6.5.2.3 Existing relations between business roles and RBAC roles

In practice, in the hospital, when a new employee is hired for the reception, the department manager assigns him to one of the eight business roles and, according to the RBAC administrator, to a set of RBAC roles. This assignment is based on, and supported by, an existing list of correspondences between both types of roles. These correspondences have as objective to facilitate the assignment of RBAC roles to the employee following the business roles they are assigned to. This correspondence between the business role and the RBAC role is equivalent to the *trace to association* illustrated in Figure 6.3. These *trace to associations* between the business roles and the RBAC roles are represented in Table 6.6.

Business roles	RBAC roles
BR1	REFRECEPTION, RR6
BR2	REFRECEPTION, RR4, RR5, RR6
BR3	REFRECEPTION, RR6, RR7
BR4	REFRECEPTION, RR6
BR5	REFRECEPTION, RR4, RR5, RR6, RR8, RR11
BR6	REFRECEPTION, RR4, RR5, RR6, RR7, RR8, RR9, RR10, RR11
BR7	RR10
BR8	RR6, RR9

Table 6.5: List of existing relations between the business roles and the RBAC roles

Summarised, the analysis of the existing access rights management in the hospital shows that, at the application layer, permissions are assigned to RBAC roles and that the latter are generated from the business roles existing at the business layer. We also observe that the assignment of the permissions to these RBAC roles has been realised progressively and empirically, and without systematic alignment with the responsibility and required rights to use business objects from the business layer. The next section analyses the responsibilities assigned to the business roles and aligns them with the permissions and the RBAC roles.

6.5.3 Analysis of the access rights really required by the business roles

In this section, we analyse the permissions which should be provided to the business roles according to the responsibilities they are assigned to, for the reception of the hospital. Therefore, we exploit the method defined in Section 6.4 which aims at performing the processes which compose the *Access right management reference model* to instantiate the *RBAC reference model*, at the application layer. In the hospital, as reviewed in previous sections, the permissions are exclusively assigned to the business roles and not to the employees, the two following processes are not considered: *Populate the list of Users* and *Populate the list of User to RBAC Roles assignments*.

In practice, in the following of this section, we perform the process *Populate the list of RBAC Roles* in Section 6.5.3.1, then we perform the process *Populate the list of Permissions* in Section 6.5.3.2, and finally, we perform the process *Populate the list of RBAC Roles to Permissions assignments* in Section 6.5.3.3.

6.5.3.1 Population of the list of RBAC roles

Given that the RBAC roles are generated from the business roles (Section 6.3), the *Populate the list of RBAC Roles* process firstly needs to analyse the business layer of the hospital to model the business roles which need to access the information system and secondly, generate the RBAC roles from these business roles.

In practice, the Human Resources (HR) department of the hospital is implied in the definition of the *Référentiel de compétences* which we translate to *Job description*. These job descriptions

aim at describing the tasks to be performed by the business role, as well as the necessary knowledge required to be assigned to it. However, the job descriptions do not specify the access rights required on professional software. Using this document, for the reception department, eight business roles have been detected and correspond to those listed in Section 6.5.2. As in this Section, we analyse the permissions really required by the business roles to compare them, in the next section, with the permissions they received and which were analysed in Section 6.5.2.2, to compare the same things, we conserve the same business roles as the one used in this Section 6.5.2.

Additionally, according to the alignment between the Responsibility metamodel and the RBAC model explained in Figure 6.3, zero to one `RBAC role trace to` one `BusinessRole`. Therefore, we have to generate one `RBAC role` to realise one `BusinessRole`. The generated RBAC roles are represented in Table 6.6 where the list of business roles is given in the first column and the RBAC roles, generated by each of these business roles, are given at the same line in the second column.

Business roles	**RBAC roles**
BR1	RR1
BR2	RR2
BR3	RR3
BR4	RR4
BR5	RR5
BR6	RR6
BR7	RR7
BR8	RR8

Table 6.6: *Trace to associations* between the business roles and the RBAC roles

6.5.3.2 Population of the list permissions

To populate the list of permissions, according to the alignment of ReMMo with the RBAC role, the first step consists in modelling the responsibilities which are assigned to the business roles, the accountabilities that are aggregated by these responsibilities and the rights to use required by these accountabilities. These responsibilities and accountabilities do not formally exist in the hospital but may be engineered from the *Job description* related to the receptionist's role analysis. Regarding the rights to use, they do not exist in the *Job description* but may be discovered by interviewing the reception manager.

To model the responsibilities, the RBAC administrator must analyse the business layer of the enterprise, and in the case of the hospital, the information provided by the *Job description*. In our case, sixteen responsibilities have been extracted for the reception. In this chapter, only the responsibility *Resp6* and *Resp14 (partially)* have been modelled with ArchiMate enhanced with the Responsibility metamodel in Figure 6.8. The other responsibilities

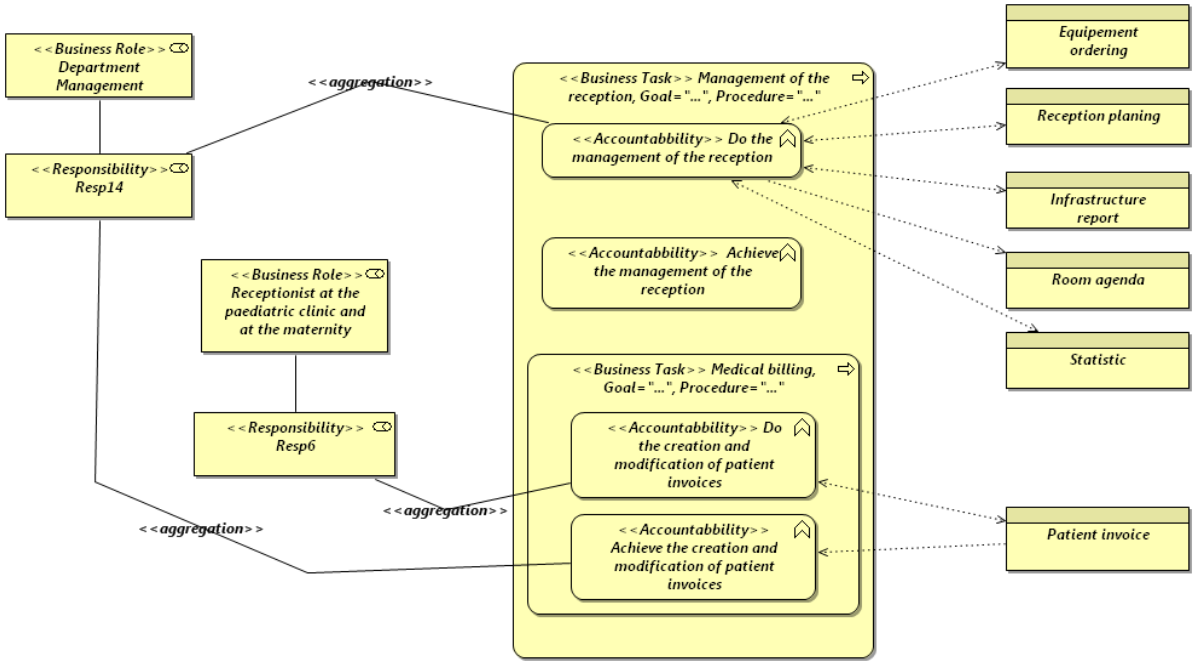


Figure 6.8: Responsibilities *Resp6* and *Resp14* (partially) modelled with ArchiMate extended with the Responsibility metamodel

are presented in Table 6.7. As illustrated on this figure, the `Departement Manager` is assigned to `Resp4` which aggregates two accountabilities, firstly, the accountability `to do the management of the reception` which requires the right to `write` the `Room agenda` and the rights to `read/write` the `Equipment ordering`, the `Reception planning`, the `Infrastructure report` and the `Statistics`, and secondly, the accountability `to achieve the creation and modification of the patient's invoices` which requires the right to `read` the `patient's invoices`. For the sake of clarity, only the accountability `to do the management of the receptionist` and the accountability `to achieve the creation and modification of patient's invoices` have been modelled in this figure. The complete list of responsibilities, accountabilities and the required rights to use is represented in Table 6.7. The first column presents the responsibilities, the second the accountabilities, and the third, the rights to use required by the accountabilities. These rights to use are afterwards described in Table 6.8.

Responsibility ID	**Accountabilities aggregated by the responsibilities**	**Rights to use required by the accountabilities**
Resp1	Do the patient entry management	RU1, RU2
Resp2	Do the patient transfer management	RU3, RU4
Resp3	Do the bed status management	RU5, RU6
Resp4	Do equipment ordering	RU7, RU8
Resp5	Do the medical delivery encoding for billing	RU9, RU10

Resp6	Do the creation and modification of patient's invoices	RU11, RU12
Resp7	Do about the bed status	RU5
Resp8	Do the realisation of the work plans	RU13, RU14
Resp9	Do the control of the monthly worksheets	RU13
Resp10	Do the management of HR indicators: Overtime, Days off, Hours of recovery	RU13, RU14
Resp11	Do the management of the room	RU17, RU18
Resp12	Do the verification of the infrastructure	RU16
Resp13	Do fix defective infrastructure	RU7, RU8, RU17
Resp14	Do the management of the reception	RU7, RU8, RU13, RU14, RU16, RU17, RU18, RU19, RU20
	Achieve the patient entry management	RU1
	Achieve the patient transfer management	RU3
	Achieve the bed status management	RU5
	Achieve equipment ordering	RU7
	Achieve the medical delivery encoding for billing	RU9
	Achieve the creation and modification of patient's invoices	RU11
	Achieve the monthly worksheets	RU13
	Achieve the management of HR indicators: Overtime, Days off, Hours of recovery	RU13
	Achieve the management of the room	RU17
	Achieve the verification of the infrastructure	RU15
	Achieve fix defective infrastructure	RU7, RU15
	Achieve inform about the doctors on duty	RU21
	Achieve the statistical analysis to follow up the daily business	RU19
Resp15	Do inform about the doctors on duty	RU21
Resp16	Do the statistical analysis to follow up the daily business	RU19, RU20

Table 6.7: List of responsibilities, accountabilities and rights to use assigned to the receptionist's business roles

RightToUse	**Rights type**	**Concerned Business objects**
RU1	Read	Basic patient's information
RU2	Write	Basic patient's information
RU3	Read	Entry, transfer or leaving patient's information
RU4	Write	Entry, transfer or leaving patient's information

RightToUse	Rights type	Concerned Business objects
RU5	Read	Bed status
RU6	Write	Bed status file
RU7	Read	Equipment ordering
RU8	Write	Equipment ordering
RU9	Read	Medical delivery
RU10	Write	Medical delivery
RU11	Read	Patient's invoices
RU12	Write	Patient's invoices
RU13	Read	Reception planning
RU14	Write	Reception planning
RU15	Read	Infrastructure report
RU16	Write	Infrastructure report
RU17	Read	Room agenda
RU18	Write	Room agenda
RU19	Read	Statistics
RU20	Write	Statistics
RU21	Read	Planning of doctors on duty

Table 6.8: List of rights to use required for the accountabilities

The second step consists of generating the permissions. According to Figure 6.3, one `Permission trace to` one `RightsToUse`. Thereby, the Tables 6.9 and 6.10 have been generated and it provides, respectively, the *trace to associations* between the `Business object` and the `Object`, and the *trace to associations* between the `RightToUse` and the `Permission`. As in the RBAC model, a `Permission` is composed of `Operation` concerning `Object`, the last two are used to express the permissions in Table 6.10.

Business objects	Objects
Basic patient's information	Basic patient's data
Entry, transfer or leaving patient's information	Entry, transfer or leaving of patient's data
Bed status	Bed status file
Equipment ordering	Data in the equipment ordering software
Medical delivery	Medical delivery data
Patient's invoices	Patient's invoices record
Reception planning	Data from the Excel file: *Timetable planning*
Infrastructure report	Data in the reporting software
Room agenda	Data in the room agenda in GroupWise multi–users software
Statistics	Data in the statistical software

Business objects	Objects
Planning of doctors on duty	Planning of doctors on duty file

Table 6.9: *Trace to associations* between the business objects and the objects

Right ToUse	Permissions	
	Operations	**Objects**
RU1	Display	Basic patient's data
RU2	Create, add, modify, delete	Basic patient's data
RU3	Display	Entry, transfer or leaving of patient's data
RU4	Create, add, modify, delete	Entry, transfer or leaving of patient's data
RU5	Display	Bed status file
RU6	Create, add, modify, delete	Bed status file
RU7	Display	Data in the equipment ordering software
RU8	Create, add, modify, delete	Data in the equipment ordering software
RU9	Display	Medical delivery data
RU10	Create, add, modify, delete	Medical delivery data
RU11	Display	Patient's invoices record
RU12	Create, add, modify	Patient's invoices record
RU13	Display	Data from the Excel file: *Timetable planning*
RU14	Create, add, modify, delete	Data in the Excel file: *Timetable planning*
RU15	Display	Data from the reporting software
RU16	Create, add, modify	Data in the reporting software
RU17	Display	Data from the room agenda in GroupWise multi–users software
RU18	Create, add, modify, delete	Data in the room agenda in GroupWise multi–users software
RU19	Display	Data in the statistical software
RU20	Create, add, modify, delete	Data in the statistical software
RU21	Display	Planning of doctors on duty file

Table 6.10: *Trace to associations* between the rights to use and the permissions

6.5.3.3 Population of the list permissions assigned to RBAC roles

To populate the list permissions assigned to RBAC roles, according to the alignment of the Responsibility metamodel and the RBAC model, the process firstly needs to model the respon-

sibilities assigned to business roles. Therefore, the RBAC administrator needs to analyse the business layer of the reception. In our case, we have analysed the *Job description* and we have interviewed the reception department manager. Based on the collected information, we have designed Table 6.11 which in the first column provides the business roles and in the second column, on the same line, the responsibilities assigned to the latter.

Business roles	**Responsibilities**
BR1	Resp1, Resp2, Resp3, Resp7
BR2	Resp1, Resp2, Resp3, Resp5, Resp6, Resp7
BR3	Resp4, Resp7, Resp15
BR4	Resp4, Resp7
BR5	Resp1, Resp2, Resp3, Resp8, Resp9, Resp10, Resp16
BR6	Resp14
BR7	Resp11, Resp16
BR8	Resp4, Resp12, Resp13

Table 6.11: List of Business roles to Responsibilities assignments

The second step consists in generating the permissions to RBAC roles assignments. According to Figure 6.3, one `Permissions-RBAC Roles Assign association trace to` one `RBR-R Assign association`. Thereby, the Table 6.12 has been generated and provides the *trace to association* between the `RBR-R Assign association` and the `Permissions-RBAC Roles Assign association`. For the sake of clarity, operations are written in **bold** and objects in *italic*.

Business roles – Responsibilities	**RBAC Roles – Permissions**
BR1 – Resp1 Resp2 Resp3 Resp7	RR1 – **Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, and data in the equipment ordering software*
BR2 – Resp1 Resp2 Resp3 Resp5 Resp6 Resp7	RR2 – **Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, and data in the equipment ordering software*, **Create, add, modify, display** *the patient's invoices record*
BR3 – Resp4 Resp7 Resp15	RR3 – **Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, and data in the equipment ordering software*, **Display** *the planning of doctors on duty file*
BR4 – Resp4 Resp7	RR4 – **Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, data in the equipment ordering software*

Business roles – Responsibilities	**RBAC Roles – Permissions**
BR5 – Resp1 Resp2 Resp3 Resp8 Resp9 Resp10 Resp16	RR5 – **Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, data in the equipment ordering software, data in the Excel file: Timetable planning, and data in the statistical software,* **Create, add, modify, display** *the patient's invoices record*
BR6 – Resp14	RR6 – **Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, data in the equipment ordering software, data in the Excel file: Timetable planning, data the room agenda in GroupWise multi–users software, and data in the statistical software,* **Create, add, modify, Display** *the patient's invoices record, the planning of doctors on duty file, and the data in the reporting software*
BR7 – Resp11 Resp16	RR7 – **Create, add, modify, display, delete** *data the room agenda in GroupWise multi–users software and data in the statistical software*
BR8 – Resp4 Resp12 Resp13	RR8 – **Create, add, modify, display, delete** *data in the equipment ordering software and data in the reporting software,* **Display** *data from the room agenda in GroupWise multi–users software*

Table 6.12: *Trace to associations* between business roles to responsibilities assignments and RBAC roles and permissions assignments

6.5.4 Analysis of the access rights actually provided compared with the access rights which are really required

In Section 6.5.2, we have collected the current permissions assigned to the business roles for the reception department of the hospital. In Section 6.5.3, we have engineered, based on the method proposed in Section 6.4, new permissions to be assigned to the identical business roles. In this section, we compare both the existing and the engineered new permissions. This comparison is provided in Table 6.13 which in the first column provides the business roles (realised by RBAC roles), the second column provides the existing permissions, the third column provides the required permissions and the last column, the difference between the existing and the engineered permissions.

Business role	**Exiting Permissions**	**Required Permissions**	**Not required Existing Permissions**
BR1	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, and data in the equipment ordering software*	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file*	**Create, add, modify, display, delete** *data in the equipment ordering software*
BR2	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, and data in the equipment ordering software*, **Create, add, modify, display** *the patient's invoices record*	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data*, **Create, add, modify** *the patient's invoices record*	**Create, add, modify, display, delete** *data in the equipment ordering software*
BR3	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, and data in the equipment ordering software*, **Display** *the planning of doctors on duty file*	**Create, add, modify, display, delete** *data in the equipment ordering software*, **Display** *the bed status file and the planning of doctors on duty file*	**Create, add, modify, display, delete** *the basic patient's data and the entry, transfer or leaving of patient's data*, **Create, add, modify, delete** *the bed status file*
BR4	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, data in the equipment ordering software*	**Create, add, modify, display, delete** *data in the equipment ordering software*, **Display** *the bed status file*	**Create, add, modify, display, delete** *the basic patient's data and the entry, transfer or leaving of patient's data*, **Create, add, modify, delete** *the bed status file*

Business role	**Exiting Permissions**	**Required Permissions**	**Not required Existing Permissions**
BR5	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, data in the equipment ordering software, data in the Excel file: Timetable planning, and data in the statistical software*, **Create, add, modify, display** *the patient's invoices record*	**Create, add, modify, display, delete** *data in the Excel file: Timetable planning, and data in the statistical software*	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, data in the equipment ordering software,* **Create, add, modify, display** *the patient's invoices record*
BR6	**Create, add, modify, display, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, data in the equipment ordering software, data in the Excel file: Timetable planning, data the room agenda in GroupWise multi–users software, and data in the statistical software,* **Create, add, modify, Display** *the patient's invoices record, the planning of doctors on duty file, and the data in the reporting software*	**Display** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data, the patient's invoices record and the planning of doctors on duty file,* **Create, add, modify, display, delete** *data in the Excel file: Timetable planning, data the room agenda in GroupWise multi–users software, and data in the statistical software, data in the equipment ordering software,* **Create, add, modify, display** *data in the reporting software*	**Create, add, modify, delete** *the basic patient's data, the entry, transfer or leaving of patient's data, the bed status file, the medical delivery data,* **Create, add, modify, display** *the patient's invoices record*
BR7	**Create, add, modify, display, delete** *data the room agenda in GroupWise multi–users software and data in the statistical software*	**Create, add, modify, display, delete** *data in the room agenda in GroupWise multi–users software and data in the statistical software*	

Business role	Exiting Permissions	Required Permissions	Not required Existing Permissions
BR8	**Create, add, modify, display, delete** *data in the equipment ordering software and data in the reporting software,* **Display** *data from the room agenda in GroupWise multi–users software*	**Create, add, modify, display, delete** *data in the equipment ordering software,* **Create, add, modify** *data in the reporting software,* **Display** *data from the room agenda in GroupWise multi–users software*	**Delete** *data in the reporting software*

Table 6.13: List of differences between existing and required rights

By comparing the access rights actually assigned to the business role (Section 6.5.2) and the ones engineered using the responsibilities (Section 6.5.3), we have observed the following not required existing permissions:

- BR3 and BR4 are assigned to too many permissions. Actually, the employees assigned to the Phone reception role and to the Infodesk role are authorised to create, add, modify, display, delete the basic patient's data and entry, transfer, or leaving data although they do not require these permissions. Additionally, they are assigned to the permissions create, add, modify and delete the bed status files although they only require the permission to display the bed status file.
- BR1, BR2, BR5 are not assigned to the responsibility which aggregates the accountability to do equipment ordering, although they have the permission to create, add, modify, display and delete data in the equipment ordering software.
- BR6 is actually assigned to all the permissions assigned to the other BR's. This was motivated by the fact that the Reception management role must supervise and monitor the other business roles. However, by analysing the responsibilities assigned to this business role, we have observed that the responsibility 14 is composed of the accountability to do the management of the reception and the accountabilities to achieve all the other business tasks of this department. In practice, the accountabilities to achieve tasks only require to read information in order to monitor the business task and not to write the information such as it is actually defined.
- BR8 is assigned the right to delete data in the reporting software although none of the responsibilities assigned to these business roles aggregates accountabilities which do not require such a permission.

Acknowledging these differences, we may conclude that using ArchiMade extended with ReMMo and the method to instantiate the *RBAC reference model* proposed by Band allows accurately defined permissions to be assigned to the RBAC role. This increase in accuracy for the definition of the required permissions is engendered by a methodical analysis of the

responsibilities and accountabilities assigned to the business role at the business layer and by systematic alignments of the concepts which composes these responsibilities with the concepts that compose RBAC.

6.5.5 Evaluation of the second part of the case study

For the second part of this case study, we had a evaluation meeting with Frank Shmitz and with Laurent Wehr. During this evaluation, Frank Schmitz evaluated our method of managing the access rights using the responsibilities. Frank Schmitz considers that our method allows obtaining results, is efficient, clear and accurate. He also considers that using the responsibility allows having traceability of the access rights that are provided to the employees and that, as a result, it is a highly accurate approach.

Frank Schmitz completely agrees that the Responsibility metamodel may be used to represent the responsibilities in the hospital, that using these responsibilities allows aligning the business processes defined at the business layer with the IT applications running at the application layer. It is, according to him, a solution that could really enhance the provisioning of the access rights to the employees and, thereby, improving the performance of the hospital.

Laurent Wehr explained the existing fuzziness in the hospital concerning the assignment of access rights to the employees. This fuzziness is due to the lack of processes and to the fact that access rights are, mostly, decided by the manager of the employee, without sufficient logic. In practice, when a new employee is hired in the hospital, its manager is urged to quickly determine the rights needed to perform the job. Afterwards, the requested rights are validated by a long chain of validations to be, *in fine*, requested to the IT department.

For Laurent Wehr, the approach based on the definition of the responsibilities sounds appropriate in the hospital's case and constitutes a suitable solution to align the access rights provided to the employees according to their real activities in a department. Laurent Wehr also analysed ReMMo and judged it in term of completeness, usefulness and understandability. Regarding the completeness, the metamodel is evaluated as complete enough to be exploited in the hospital, it is useful for the provisioning of the access rights and its usage could be extended to other domains, such as the formal representation of the activities to be performed by the employees or the management of their competences. Laurent Wehr has easily understood the Responsibility metamodel, although he was not familiar with the UML formalism. However, he cautions to be careful regarding the understanding of this language, for further communications with persons from other businesses.

6.6 Conclusions

In Chapter 5, we have mapped the Responsibility metamodel with ArchiMate. This mapping has allowed to model the responsibility using the enterprise architecture metamodel and thereby, enhancing the alignment between the concepts modelled at the business layer. In this chapter, first we have proposed an alignment between ReMMo and the Role Based Access Control model. This alignment has resulted, amongst others, in the definition of a set of *trace to associations* between the concepts from both latter such as the `User trace to Employee`, the `RBAC Role`

`trace to Business Role`, the `Permission trace to Responsibility`, the `RightToUs trace to the Operation` and the `BusinessObject trace to the Object`. Secondly, we have proposed a method to populate, based on this alignment, the *RBAC reference model* existing at the application layer.

Afterwards, to illustrate and evaluate this alignment and the method, we have introduced the second part of the case study in the Centre Hospitalier de Luxembourg. This case study aimed at improving the alignment of the business layer of the hospital with its application layer. Therefore, at the business layer, the human resources department of the hospital defines *Job descriptions* which formalise the responsibilities which are to be achieved by the business roles. The case study has, successively, presented (1) the existing business layer and the application layer components, as well as the permissions actually provided to the existing business roles, (2) an analysis, based on the responsibilities and using our method, of the real required permissions to be provided to the business roles and (3) a comparison of the existing permissions against the permissions really required based on the responsibilities. The results of this comparison were that seven business roles over eight are assigned with too many permissions.

Finally, we evaluate this case study with Frank Schmitz, Reception department manager, and with Laurent Wehr, Manager of the competences management. Both of them having participated in the realisation of this case study.

Publications related to this chapter:

- C. Feltus, M. Petit, M. Sloman, Enhancement of Business IT Alignment by Including Responsibility Components in RBAC, in *Proceedings of the 5th International Workshop on Business/IT Alignment and Interoperability (BUSITAL)*, Hammamet, Tunisia. 2010.
- M. Petit, C. Feltus, F. Vernadat, Enterprise Architecture Enhanced with Responsibility to Manage Access Rights – Case Study in an EU Institution, in *Proceedings of The Practice of Enterprise Modeling – 5th IFIP WG 8.1 Working Conference (PoEM)*, Rostock, Germany. 2012.
- C. Feltus, E. Dubois, E. Proper, I. Band, M. Petit, Enhancing the ArchiMate® Standard with a Responsibility Modeling Language for Access Rights Management, in *Proceedings of the 5th International Conference on Security of Information and Networks (SIN)*, Jaipur, Rajastan, India. 2012. ACM.

Chapter 7

Conclusions

To conclude this thesis, in Section 7.1, we first summarise the successive results of this thesis. Then, in Section 7.2, we evaluate the research according to Hevner's guidelines. Finally, in Section 7.3, we provide perspectives for future works.

7.1 Summary of the research

This thesis has been introduced, in Chapter 1, by the statement that, nowadays the economy relies on companies operating sometimes with an information system shared by thousands of employees, continuously evolving, and gaining more and more flexibility and openness. Using these sophisticated information systems, companies possess powerful tools to manage all the dimensions of their business, from the management of customers, production activities, stocks, or human resources. In such a complex and evolving environment, aligning the business down to the appropriate IT infrastructure is a challenging activity that needs to be carefully handled. The alignment of the application layer with the business layer is critical for the companies who, without such an alignment, risk not to be able to deliver their services any more and thus risk to be seriously discredited by their customers, thereby, jeopardising their image. One aspect of this business/IT alignment concerns the alignment of the access rights to data and applications required by the employees according to the information they need to perform business activities. In this area, our state of the art of the access rights models and engineering methods, in Chapter 2, has highlighted an evolution towards more consideration of business concepts. Indeed, the access rights management solutions gradually progresses towards a wider integration of the business concepts such as employees' obligations and responsibilities regarding the tasks they are assigned to.

In parallel, many governance standards and norms have acknowledged the above alignment challenges and have highlighted new needs to be fulfilled in terms of business/IT alignment and access rights management. In Chapter 3, the review of these standards and norms has allowed to draw an unrefined picture of the zones of concepts to be taken into account when addressing governance requirements. Unfortunately, we have observed that the access rights management solutions do not yet fully consider and integrate these concepts. For the moment, there exists

no solution allowing thoroughly connecting the business layer and application layers and, alike, no common model is yet agreed upon among business and the IT staff, especially concerning the management of the access rights. Despite our observation related to the need for considering the concept of responsibility, as well as a set of concepts that allow defining it, such as the accountability, the capability or the right to use, up to date no approach really considers these concepts, as reviewed in Chapter 2. Hence, this observation has led us to the first research question of the thesis, which is, *Considering the corporate and IT governance needs, what are the concepts which constitute the core of the employee responsibility and how these concepts may be associated in a dedicated Responsibility metamodel?* To answer the question, we have analysed the literature from the field of IS/IT and from the field of the human sciences. The literature analysed has allowed us to discover the amplitude of the notion of responsibility that gathers, at the same time, information related to (1) the accountabilities of the employees, which are mainly defined at the business layer. (2) the rights and capabilities that these employees require to perform their accountabilities. These rights and capabilities are issued from the business layer but impact the application layer, with e.g. the definition of the access rights. And (3) the assignment of responsibilities to employees, directly or through the roles they play. Knowing the meaning and acknowledging the importance of these concepts led us to the definition of the Responsibility metamodel, in Chapter 4. This Responsibility metamodel allows the defining of responsibilities at the business layer and, thereby, allows engineering the access rights required to perform the responsibilities, to be provisioned at the application layer.

Enterprise architecture modelling is based on approaches which enable illustrating the inter relations between the different layers of a company and between the different aspects that it addresses such as the behaviour, the information, or the people. Enterprise architecture metamodels provide views which are understandable by all the stakeholders and which allow making decisions knowing the impact over the company. However, the problem with the enterprise architecture metamodels is that, in general, the concepts which compose them lack precision and, therefore, are hardly usable to perform, verify or justify concrete alignments. Acknowledging this statement, we concluded that, in practice, enterprise architecture metamodels do not permit accurate engineering of the access rights to be provisioned to the employees at the application layer, based on the specification from the business layer such as required by the governance standards and norms. In parallel, we also considered that the enterprise architecture metamodels provide a good basis for this since they model the most significant concepts related to the information system of a company. To reap the benefits of the enterprise architecture metamodel for the engineering and the management of the access rights, we have decided to focus our research on integrating ReMMo with ArchiMate, as an example of enterprise architecture modelling language, and we proposed the following second research question: *How may business/IT alignment be improved considering the responsibility, in the context of enterprise architecture metamodel, and for the field of access rights management?* In order to answer this question, we have integrated the Responsibility metamodel with the business layer of the enterprise architecture metamodel, in Chapter 5. After this integration, the associations between the business concepts of business actor, business role, business process, business function and business object have been semantically enriched. The answer to the second research question is provided by an integrated Responsibility and ArchiMate metamodel. This integrated metamodel allows refining the responsibilities and accountabilities of employees and assigning access rights to them considering the accountabilities which compose the responsibilities.

Using the concept of role for the management of access rights is an approach commonly

agreed upon by most of the companies. RBAC is the leading model in this area and is based on two processes: the assignment of users to roles and the assignment of permissions to roles. RBAC is a model that allows optimising, at the application layer, the assignment of a large number of permissions to a large number of roles. Throughout the literature, we have accordingly observed that the RBAC role is mostly considered as a business role. In practice, however, we see that the concept of role is used at the application layer, where application roles are exploited. This was, for instance, the case in our case study at the hospital. The non–alignment between the business roles and the RBAC roles led us to the research sub–question : *How may responsibility be mapped with the role based access control model and how does this mapping enhance the accuracy and the usability of the access rights management?* To answer this question, In Chapter 6, we have first aligned our Responsibility metamodel with RBAC. This alignment has allowed the tracing of the relationships, amongst others, between the user and the employee, between the RBAC role and the business role, and between the permission and the responsibility. Secondly, we have proposed an *Access rights management reference model*, at the business layer of ArchiMate. This reference model has been designed in a way that it may be supported, at the application layer, by RBAC based solution, and hence, by the Band's reference model. In practice, the alignment of ArchiMate extended with the Responsibility metamodel with RBAC, and the processes modelled in the *Access rights management reference model*, constitutes our method for the engineering of access rights management based on the employees' responsibility. In this method, the concept of responsibility is used as a pivot between the business layer and the application layer. It offers the advantage to integrate the requirements from both layers, namely:

- on the first hand, at the business layer, employees are gathered in business roles and those business roles are organised in an organisational chart and are assigned to a set of responsibilities which concern a precise business task;
- on the other hand, at the application layer, the assignment of access rights to the employees is optimised using the concept of RBAC role which gathers all the permissions required by a business role.

In summary, answering these research questions has allowed enhancing the alignment between the business layer and the application layer using the employees' responsibilities, such as required by the governance needs. It has also improved the expressiveness of the enterprise architecture metamodels, and, as a result, their function of supporting the business/IT alignment, and more specifically the management of access rights.

To illustrate our approach, the integration of ReMMo with ArchiMate has been used in a real life case study performed in two parts at the Centre Hospitalier de Luxembourg. In the first part, responsibilities have been defined for the activity of providing accesses to medical patient records based on a set of scenarii and according to the roles played by the employees. In the second part, responsibilities of the employees of the reception department have also been defined, and have been mapped with two concepts, equivalent to RBAC roles at the application layer: the *Reference User* and the *Functional Role*. Both parts of the case study have shown the added value of the Responsibility metamodel mapped with ArchiMate: the first part for defining the responsibilities, and the second part, based on the alignment of ArchiMate extended with the responsibility with RBAC, to evaluate the relevance conferred by our approach to define access rights.

A complementary case study also took place at the European Court of Auditors were ReMMo has been integrated in the ECA enterprise architecture metamodel, as explained in Appendix

G. Using that integrated metamodel, we have enhanced the user provisioning and account management process and we have defined the responsibilities of all the employees involved in this process. This work has allowed precisely defining the responsibilities, including the accountabilities, and the required capabilities and rights. Thereby, it has also evaluated the applicability of the approach to another enterprise architecture framework: the ECA enterprise architecture metamodel.

Finally, the work realised during the case studies at the hospital and the court with the protagonists from both institutions has been presented and evaluated. Over all, people believe that using the responsibility is an interesting idea which is unanimously considered as useful to improve the performance of the companies. They also generally believe that the metamodel of responsibility is understandable, complete and accurate, and that the mapping of this Responsibility metamodel with enterprise architecture frameworks is an appropriate and useful approach to align the business layer with the application layer. Additionally, some of them emphasised the precision and rigour of the approach and, the advantage in terms of the traceability that it confers.

7.2 Evaluation of the research according to Hevner's guidelines

As explained as introduction of this thesis, Hevner et al. (2004) propose a model to appraise research achieved in the field of design science using seven guidelines. In this section, we review each guideline and evaluate our research, accordingly.

7.2.1 Guideline 1: Design as an artefact

In design science, an artefact is characterised by the importance of the organisational problem that it addresses and represents the output of build activities, namely: concept, model, method or instantiation. These artefacts are considered as innovations which define new elements (e.g., ideas, products, and so forth) and contribute to enhance the development and the exploitation of information systems.

The thesis has contributed to the design of a set of artefacts as summarised in Chapter 1, Tables 1.1 and 1.2. The artefacts are the Responsibility metamodel, its integration in enterprise architecture frameworks and its alignment with RBAC, and an access rights management method. The latter enhances the alignment between the business layer and the application layer. These artefacts contribute in solving an important organisational problem in the field of business/IT alignment and in the field of access rights management. Moreover, as expressed in the preamble of Chapter 4, the objective of the Responsibility metamodel is to model exclusively the responsibilities related to the realisation of business tasks. This fact constitutes a limitation to the research.

7.2.2 Guideline 2: Problem relevance

The second guideline that structures design science requires that the problem solved by the research must be an "important and relevant business problem". By this statement, Hevner et al.

(2004) advocate that a dedicated community must recognise the problem and must acknowledge the relevance of the research. Accordingly, this community includes both: the community that is concerned by the information system and the community that is concerned by the development and implementation of this information system.

In Chapter 2, we have reviewed the state of the art in access rights models and roles engineering methods. This review has highlighted an evolution of the models and methods towards a deeper consideration for the business layer. In Chapter 3, we have observed that despite this, the application layer and the business layer do not completely share a common understanding of the business concepts such as responsibility and accountability. This non–alignment appears hence as a relevant and recognised problem in terms of IT governance.

In the real world, on a professional level, we have noted in Section 1.4, that the lack of consideration of the governance's needs during the engineering of the access rights leads to the additional following problems: (i) insufficient analysis of the business role, (ii) situations where the employees are assigned to too many or not enough rights regarding their responsibilities, and (iii) misalignment between the business role and the application role.

7.2.3 Guideline 3: Design evaluation

The design evaluation of an artefact involves its assessment within appropriate business settings and using appropriate metrics. In the design process, the elaboration of the artefact is based on an iterative approach. As a consequence, the evaluation of the artefact serves, afterwards, to improve this artefact elaboration.

The research has been subject to a partial evaluation, due to the fact that we have not used dedicated evaluation metrics. This partial evaluation has been performed by means of case studies. The first part of the case study in the hospital has evaluated the expressiveness of ReMMo. The second part has evaluated, firstly, that the integrated ArchiMate and Responsibility metamodel, at the business layer, enhances the usability of the definition of the access rights required by business roles, and, secondly, that the definition of the responsibilities at the business layer may be exploited to accurately generate the RBAC roles and permissions. The evaluations of the research by professionals, realised in the hospital in Sections 5.5.5 and 6.5.5, and at the European Court of Auditors in Appendix G6, have allowed evaluating the expressiveness of the Responsibility metamodel and that its integration with ArchiMate and its alignment with RBAC enhance the generation of permissions and their assignments to employees.

7.2.4 Guideline 4: Research contribution

A design science research must, according to guideline 4, provides a research contribution that either takes the form of a design artefact that allows solving a sound business problem, a foundation that is an artefact that contributes to extend the design science knowledge base, or a methodology that also contributes to enhance this knowledge base.

The evaluation of the contribution of a design research, as advised by Hevner et al., is realised by analysing the implementability of the designed artefact. The Responsibility metamodel is one main artefact designed through the research. This metamodel extends the knowledge base

related to the management of the access rights given that it enhances the definition of the rights required by the actors assigned to precise responsibilities. The integration of this Responsibility metamodel and ArchiMate is a result which, on his side, contributes to improve the knowledge base related to the alignment among the ArchiMate elements, in order to support the definition of the access rights provided to the employees according to their responsibilities. Finally, the method for managing the access rights based on the mapping of RBAC with ReMMo is a contribution to the alignment between the business layer and the application layer, and, thereby, to the state of the art related to the rights and roles engineering methods. The implementation of the research artefacts has been assessed with the professional and, in the hospital, has permitted to more accurately define the access rights to be provided to the employees. Accordingly, the evaluation of the research contribution has shown that sound business problems may be solved using the responsibility.

7.2.5 Guideline 5: Research rigour

The research rigour is an important aspect of the design science and rigorous methods need to be used during the design of the artefacts and during their evaluations. However, Hevner et al. advise not to reduce the relevance of the artefacts at the profit of the rigour. Considering this, in some cases, too much mathematical formalism may decrease the applicability of these artefacts. In order to be rigorous, Hevner et al. advise to use and exploit methods and data from approved theories and behavioural science.

To construct our first artefact, namely the Responsibility metamodel, we have used recognised theories from human, social, administrative and management sciences. To integrate the Responsibility metamodel with ArchiMate, and the alignment with RBAC, we have systematically analysed the correspondence between concepts, and associations between concepts, according to the methods proposed by Zivkovic et al. (2007) and Parent and Spaccapietra (2000). Regarding the mapping of ReMMo with the ECA metamodel, we have exploited the method proposed by Petit (2003), itself inspired from Parent and Spaccapietra (2000).

7.2.6 Guideline 6: Design as a search process

Research is a process, in design science, that tends to discover the optimal solution to solve a relevant problem, using the appropriate means, and considering the laws which structure and constrain the problem environment. Iterative approaches are often necessary to reach this solution. Most of the time, researchers begin with the analysis of a sub–problem that simplifies or decomposes the main one, and they expand the scope of the research all along the progression of their work. Notwithstanding the solution elaboration, in the mean time, the research must continuously analyse the approach feasibility along the design steps. Having the solution always appropriate according to the problem is a basic requirement.

Guideline 6 is more a statement than a guideline. However, we acknowledge that the elaboration of our solution to improve the business/IT alignment and the definition of the access rights for the employees has been achieved according to a well defined progression. Firstly we have analysed the state of the art related to the access rights management and the governance needs. This first step has clearly allowed us to figure out the problem to be addressed. Secondly, a set of significant concepts to define the Responsibility metamodel has been reviewed.

These concepts have been analysed, and their description improved, through the review of the literature. Thirdly, we have gathered and associated them in a Responsibility metamodel. This activity has been iteratively enhanced according to the integration of new concepts, following its confrontation with professional frameworks (e.g., with COBIT), and through its deployment during case studies (e.g., the intermediary case study at the European Court of Auditors). Finally, the metamodel of Responsibility has been integrated with ArchiMate and aligned with RBAC.

7.2.7 Guideline 7: Communication of research

As the design science objective is to solve a real problem, the solution proposed must, on the one hand, be appropriately transferred to the practitioner of the concerned field. This means that the latter must be able to take advantage of it and must be aware of the knowledge required to apply it. On the other hand, communication must also be realised with the researchers to enable them to enhance their knowledge database. In the latter case, the communication concerns the designed artefacts as well as the method used to elaborate them. Hevner at al. also emphasises the importance of having the communication adapted according to the profile of both targets. The authors advise, for instance, not to describe some details when transferring the artefacts to the practitioners.

During this research, communication has been realised towards the scientific community and towards the practitioners as well. Towards the practitioners, we have communicated the research output, mainly, during the two case studies where the Responsibility metamodel was tested with the existing infrastructure. The communication towards practitioners also occurred during the presentation of the integration of ArchiMate with ReMMo at The Open Group conference, as explained in the next section.

The communication towards the researchers has mainly been realised by the intermediary of scientific publications. The eleven publications realised have been presented in Section 1.10.

7.3 Future works

We retain four opportunities for future works. The first one consists in using the responsibility to improve the engineering of alternative access rights management solutions, in Section 7.3.1. The second one consists in analysing the mutability of the Responsibility metamodel and access rights management method, in Section 7.3.2. The third interesting future work consists in a more in depth analysis of the integration of the responsibility in ArchiMate, by exploring the development of the capability concept available in this framework, in Section 7.3.3. Finally, the fourth one concerns the development of a possible tool for supporting the usability of the access rights management method, in Section 7.3.4.

7.3.1 Alternative access rights management solutions

Future works related to the Responsibility metamodel could concern the elaboration of alternative access rights management in two fields: the services systems and the critical infrastructures.

Service system (Maglio and Spohrer (2008)) is an important field of research which aims, at addressing and reducing the complexity of information systems by integrating different disciplines such as, e.g., computer science, economics, human resources management, or cognitive science, and which provides new cooperation opportunities for companies. Hence, a service offers a well settled and easy way for sharing data, applications or knowledge in a precise enclosed space of the IS. Notwithstanding the obvious benefits offered, service systems still suffer from engineering difficulties which need to be focused on. Among the glaring challenges to consider, the service sharing and interoperability appear a crucial issue. Therefore the first future work could aim at introducing the responsibility as an integral part of the services modelling, in order to sustain the interoperability between the access rights management solutions of these services systems.

Nowadays, critical IT infrastructures constitute the pillars of our economy and require dynamic protection mechanism. In this context, being able to quickly react and in real time is a crucial challenge for the security officers in charge of maintaining those infrastructures operational and thus avoiding a crisis. Many architectures exist to dynamically support the reaction after the detection of an incident. These architectures are mostly elaborated based on a multi–agents systems approach which offers the possibility to work in decentralised and heterogeneous environments. Despite the evolution of the existing solutions towards more dynamism, we have observed that these architectures are based on static assignments of functions to agents and that, as a result, isolating an agent or breaking the communication channel between two of them could create serious damage on the management of the crisis. In this second future work, we propose to address an innovative approach for making the assignment of functions to agents in the critical architecture more dynamic. Therefore, our future work aims at exploiting the notion of agent's responsibility which ought to be assigned dynamically to agents depending on the agent's capabilities, and thus on the type and severity of the crisis. Simultaneously, dynamic assignments of the necessary access rights to perform the accountabilities which compose the agent's new responsibilities could also be possible, which could realise the protection of the critical infrastructure more quickly.

7.3.2 Mutability of the Responsibility metamodel and access rights management method

As explained in Section 1.6, the scope of our research has been limited to highly regulated enterprises, of a type "bureaucratic". A specificity of these companies is that their organisation is very rigid. Their processes, roles, functions, tasks, hierarchy, access rights and so forth are defined "by design" and there exists no room for dynamic and exceptional modifications like, for instance, the delegation of a task which has not been intentionally and deliberately foreseen, beforehand. Future works regarding this specificity of the access rights management could consist in analysing how the Responsibility metamodel, and the access rights management method, could be extended and completed to address the needs of providing access rights to the employees "on the fly", while staying compliant with the governance needs review in Chapter 3.

In Chapter 6, the Responsibility metamodel has been aligned with RBAC to provide the access rights to the employees considering this specific access control model. This alignment with RBAC was motivated by the level of consideration and deployment of the model through the scientific and professional community. Despite this level of integration of RBAC, we have also

noted in our state of the art related to the access control models that other models exist and are deployed through professional solutions and systems (e.g.1, MAC is implemented in UBUNTU, FreeBSD or SUSE. E.g.2, DAC is implemented in Unix, Mac OS X version 10.4 or the IBM AS/400 technology). Considering this, an interesting future work remains in aligning the Responsibility metamodel with these other significant models such as MAC, DAC, OrBAC, TBAC, and so forth. The advantage of such an alignment with other access control models is that, based on the employee responsibilities, it is also possible, at the same time, to populate different types of access control models, spread on different applications, without any supplementary rights engineering activities and methods.

7.3.3 Contribution to ArchiMate

The third future direction consists of pursuing the integration of the Responsibility metamodel with ArchiMate. At this level, one future work is related to the exploration and the definition of the concept of capability within the framework. This exploration and definition has already been initiated by some researchers, as explained in Section 5.4.5, and it could be complemented and enriched with the definition of the capability from this thesis. A second future work consists of pursuing the work related to the integration of the responsibility in ArchiMate and with RBAC. In January 2012, a working group has been initiated by The Open Group at the San Francisco conference. The expected result of the new working group is a white paper addressing the integration of security concepts in ArchiMate. During this working group, the integration of the Responsibility metamodel in ArchiMate has been presented, among others, to the following experts in IT security and enterprise architecture: Dr Henry Franken[1], Ian Dobson[2], Jim Hietala[3] and Iver Band[4]. For them, the Responsibility metamodel is complete and suitable to positively enhance the performance of the companies. They consider that the Responsibility metamodel could contribute in aligning the business layer with the application layer, and that using the concept of responsibility is appropriate to provision the employees with the access rights they require to perform business activities. Considering these comments, this thesis dissertation could also be a potential contribution to the working group.

7.3.4 Enhancement of the usability

The final future work could aim at strengthening the usability of the Responsibility metamodel and access rights management method. This usability has been addressed, in Chapter 5 by integrating the Responsibility metamodel with the business layer of ArchiMate, to benefit from the ArchiMate language formalism.

A first future work to sustain this usability is related to the enhancement of the connections between the strategic and operational layers of the company, and, thereby, to the support of the decision making for the managers. Therefore, the Responsibility metamodel could be integrated with frameworks that also address the strategic layer, such as i* (Yu (1996)) or the "Design and Engineering Methodology for Organisations" (DEMO – Dietz (2001)). DEMO provides a

[1]Owner of BiZZdesign, http://www.bizzdesign.com/

[2]Director of The Open Group's Security Forum since February 2001

[3]Vice President Security for The Open Group

[4]Enterprise Infrastructure Architect at Standard Insurance Company (http://www3.standard.com/) and author of the *RBAC reference model* used in Chapter 6

methodology for modelling and designing organisations, as well as the IT which supports the latter. Therefore, DEMO considers that the organisation may be modelled following three levels of abstraction: business, information and data systems. Following these layers, a set of models are defined, and a set of diagrams are engineered (i.e.: communication diagram, process diagram, transaction diagram, fact diagram, and action diagram). The DEMO method is based on the analysis of these diagrams, gathered in business processes, following a transaction perspective. Therefore, it considers that, in a sufficiently precise diagram, actors have the ability and the responsibility to perform actions regarding a dedicated transaction. Practically, this future work should contribute in strengthening the definition of these responsibilities using the Responsibility metamodel and their modelling with DEMO.

A second future work related to the enhancement of the usability concerns the development of a tool to support the deployment of the access rights management method defined in Chapter 6. In practice, this tool should act as a work–flow which populates the *RBAC reference model* according to the alignment of the Responsibility metamodel and RBAC, and following the five processes described in Section 6.4, respectively related to the population of the list of users, the list of RBAC roles, the list of permissions, the list of users to RBAC roles assignments and the list of RBAC roles to permissions assignments. Given the tool's objective, close connections with enterprise modelling tools (e.g., Archi[1]) should be established and, as a result, new representations of the concepts related to the responsibility designed.

7.4 Publications related to future works

- C. Bonhomme, C. Feltus, M. Petit, Dynamic Responsibilities Assignment in Critical Electronic Institutions – A Context–Aware Solution for in Crisis Access Right Management, in *Proceedings of the 6th International Conference on Availability, Reliability and Security (ARES)*, Vienna, Austria. 2011. IEEE.

- C. Feltus, A. Khadraoui, A. Yurchyshyna, M. Léonard, E. Dubois, *Responsibility aspects in service engineering for eGovernment*, in *Proceedings of the Workshop of the 6th Interoperability for Enterprise Systems and Applications conference (I–ESA), Service Science and the next wave in Enterprise Interoperability*, Valencia, Spain. 2012.

- A. Khadraoui, C. Feltus, Services Specification and Services Compliance, How to Consider the Responsibility Dimension?, in *Journal of Service Science Research – "Challenges and Advances on Service Quality Aspects"*, Springer, volume 4, issue 1, pp. 123–142.

- G. Guemkam, C. Feltus, C. Bonhomme, P. Schmitt, D. Khadraoui, Z. Guessoum, Reputation based Dynamic Responsibility to Agent for Critical Infrastructure, in *Proceedings of the International Conference on Intelligent Agent Technology*, Lyon, France. 2011. IEEE, ACM.

[1] Archi is a free and open source modelling tool to create and work with ArchiMate models and sketches – http://www.archimatetool.com/

Bibliography

John M. Ackerman. Social accountability in the public sector: A conceptual discussion. 2005. 84, 85, 87, 88, 90

Gail-Joon Ahn and Ravi Sandhu. Role–based authorization constraints specification. *ACM Trans. Inf. Syst. Secur.*, 3(4):207–226, November 2000. ISSN 1094-9224. doi: 10.1145/382912.382913. URL http://doi.acm.org/10.1145/382912.382913. 27

Daniel Amyot, Jennifer Horkoff, Daniel Gross, and Gunter Mussbacher. A lightweight grl profile for i* modeling. In *Proceedings of the ER 2009 Workshops (CoMoL, ETheCoM, FP–UML, MOST–ONISW, QoIS, RIGiM, SeCoGIS) on Advances in Conceptual Modeling – Challenging Perspectives*, ER '09, pages 254–264, Berlin, Heidelberg, 2009. Springer–Verlag. ISBN 978-3-642-04946-0. doi: 10.1007/978-3-642-04947-7_31. URL http://dx.doi.org/10.1007/978-3-642-04947-7_31. 76, 88

Anne Anderson. Xacml profile for role based access control (rbac). Technical Report Draft 1, OASIS, February 2004. 152

Guillermo Navarro Arribas. Access conrol and mobile agents, 2003. Submitted to Universitat Autònoma de Barcelona in partial fulfilment of the requirements for the degree of Master of Computer Science, Supervised by Dr. Joan Borrell. 22

AS8015. 2005. AS8015:2005, The Australian Standard for Corporate Governance of ICT. 53

Vijayalakshmi Atluri and Janice Warner. Supporting conditional delegation in secure workflow management systems. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, SACMAT '05, pages 49–58, New York, NY, USA, 2005. ACM. ISBN 1-59593-045-0. doi: 10.1145/1063979.1063990. URL http://doi.acm.org/10.1145/1063979.1063990. 76

Fabien Autrel, Frédéric Cuppens, Nora Cuppens-Boulahia, and Céline Coma-Brebel. MotOrBAC 2: a security policy tool. In *SARSSI'08 : 3ème conférence sur la Sécurité des Architectures Réseaux et des Systèmes d'Information, 13–17 octobre, Loctudy, France*, 2008. 31

Loyd Baker, Paul Clemente, Bob Cohen, Larry Permenter, Byron Purves, and Pete Salmon. Foundational concepts for model driven system design. *INCOSE Model Driven System Design Interest Group*, 2000. 16

Robert W. Baldwin. Naming and grouping privileges to simplify security management in large databases. volume 0, page 116, Los Alamitos, CA, USA, 1990. IEEE Computer Society. doi: http://doi.ieeecomputersociety.org/10.1109/RISP.1990.63844. 25

BIBLIOGRAPHY

Iver Band. Modeling rbac with sabsa, togaf and archimate, creating a foundation for understanding and action. In *The Open Group Conference*, Austin, Texas, 2011. xx, 19, 148, 149, 150, 151

Alistair Barros, Keith Duddy, Michael Lawley, Zoran Milosevic, Kerry Raymond, and Andrew Wood. Processes, roles, and events: Uml concepts for enterprise architecture. In *UML 2000 – The Unified Modeling Language, Advancing the Standard, Third International Conference*, pages 62–77. Springer. 7

Basel2. 2004. Basel Committee on Banking Supervision. (June 2004). International convergence of capital measurement and capital standards; A revised framework. Bank for International Settlements. xix, 3, 10, 56, 57, 60, 61

Jörg Becker, Philipp Bergener, Patrick Delfmann, Mathias Eggert, and Burkhard Weiß. Supporting business process compliance in financial institutions – a model–driven approach. In *Wirtschaftsinformatik*, page 75, 2011. 79

Daniel Beimborn, Frank Schlosser, and Tim Weitzel. Proposing a theoretical model for it governance and it business alignment. *Hawaii International Conference on System Sciences*, 0:1–11, 2009. doi: http://doi.IEEEcomputersociety.org/10.1109/HICSS.2009.873. 48

E. D. Bell and J. L. La Padula. Secure computer system: Unified exposition and multics interpretation, 1976. URL http://csrc.nist.gov/publications/history/bell76.pdf. 22

Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role–based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, August 2001. ISSN 1094-9224. doi: 10.1145/501978.501979. URL http://doi.acm.org/10.1145/501978.501979. 29, 75, 79

Ranjita Bhagwan, Fred Douglis, Kirsten Hildrum, Jeffrey O. Kephart, and William E. Walsh. Time–varying management of data storage. In *Proceedings of the First conference on Hot topics in system dependability*, HotDep'05, pages 14–14, Berkeley, CA, USA, 2005. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1973400.1973414. 1

Biba. Integrity Considerations for Secure Computer Systems. *MITRE Co., technical report ESD–TR 76-372*, 1977. 23

Ken Birman, Gregory Chockler, and Robbert van Renesse. A structured taxonomy of responsibility concepts. *SIGACT News*, 40:68–80, June 2009. ISSN 0163-5700. doi: http://doi.acm.org/10.1145/1556154.1556172. URL http://doi.acm.org/10.1145/1556154.1556172. 1

Thomas Bivins. *Responsibility and Accountability, in Ethics in Public Relation: Responsible Advocacy*, chapter 2, pages 19–38. CA:Sage, 2006. 68, 88

Peride K. Blind. Accountability in public service delivery: A multidisciplinary review of the concept. *Prepared for the Expert Group Meeting Engaging Citizens to Enhance Public Sector Accountability and Prevent Corruption in the Delivery of Public Services*, 2011. 66, 67, 68, 84, 85, 88

G. Blokdijk and I. Menken. *The Service Level Agreement SLA Guide: SLA Book, Templates for Service Level Management and Service Level Agreement Forms. Fast and Easy Way to Write Your SLA*. Emereo Pty Limited, 2008. ISBN 9781921523625. URL http://books.google.be/books?id=5faVvaHSZfoC. 77, 78, 225

A. J. C. Blyth, J. Chudge, J. E. Dobson, and M. R. Strens. Ordit: a new methodology to assist in the process of eliciting and modelling organizational requirements. In *Proceedings of the conference on Organizational computing systems*, COCS '93, pages 216–227, New York, NY, USA, 1993. ACM. ISBN 0-89791-627-1. 71, 87

Mark Bovens. Analysing and Assessing Accountability: A Conceptual Framework. *European Law Journal*, 13(4):447–468, July 2007. ISSN 1351-5993. doi: 10.1111/j.1468-0386.2007.00378.x. URL http://dx.doi.org/10.1111/j.1468-0386.2007.00378.x. 67, 85, 87, 88, 90

Mark Bovens. Two concepts of accountability: Accountability as a virtue and as a mechanism. *West European Politics*, 33(5):946–967, 2010. 66, 67, 68, 84, 85, 88

D. F. C. Brewer and M. J. Nash. The chinese wall security policy. In *IEEE Symposium on Security and Privacy*, pages 206–214, 1989a. 92

David F. C. Brewer and Micheal J. Nash. The chinese wall security policy. *Security and Privacy, IEEE Symposium on*, 0:206, 1989b. ISSN 1540-7993. doi: http://doi.IEEEcomputersociety.org/10.1109/SECPRI.1989.36295. 25

M. Broadbent. Creating effective it governance. In *Gartner Symposium ITEXPO*, pages 1–20. Gartner, 2002. 47

Giorgio Bruno and Marco Torchiano. Process enabled information systems. In *ICEIS*, pages 32–37, 2000. 79

Susanne Busse, Ralf-Detlef Kutsche, Ulf Leser, and Herbert Weber. Federated information systems: Concepts, terminology and architectures. Technical report, Technische Universität Berlin, Fachbereich 13 Informatik, 1999. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.5830. 114

Artur Caetano, Antonio Ritó Silva, and José Tribolet. Using roles and business objects to model and understand business processes. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1308–1313, New York, NY, USA, 2005. ACM. ISBN 1-58113-964-0. doi: 10.1145/1066677.1066973. URL http://doi.acm.org/10.1145/1066677.1066973. 81

A. Cenys, A. Normantas, and L. Radvilavicius. Designing role–based access control policies with uml. *Journal of Engineering Science and Technology Review*, 2(1), 2009. 149, 150

Yolande E. Chan and Blaize Horner Reich. It alignment: what have we learned? *JIT*, 22(4): 297–315, 2007. 48

R. Chandramouli. A framework for multiple authorization types in a healthcare application system. In *Proceedings of the 17th Annual Computer Security Applications Conference*, ACSAC '01, pages 137–, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1405-7. URL http://dl.acm.org/citation.cfm?id=872016.872160. 36

Laurence Cholvy, Frédéric Cuppens, and Claire Saurel. Towards a logical formalization of responsibility. In *ICAIL '97: Proceedings of the 6th international conference on Artificial intelligence and law*, pages 233–242, New York, NY, USA, 1997. ACM. ISBN 0-89791-924-6. doi: http://doi.acm.org/10.1145/261618.261658. 69, 84, 85, 86, 87

David D. Clark and David R. Wilson. A comparison of commercial and military computer security policies. *Security and Privacy, IEEE Symposium on*, 0:184, 1987. ISSN 1540-7993. doi: http://doi.IEEEcomputersociety.org/10.1109/SP.1987.10001. 25, 27

Eli Cohen. Reconceptualizing information systems as a field of the discipline informing science: from ugly duckling to swan. *Journal of Computing and Information Technology*, 7(3):213–219, 1999. 75

COSO. Enterprise Risk Management – Integrated Framework (Executive Summary), September 2004. The Committee of Sponsoring Organizations of the Treadway Commission. 3

Michael Covington and Manoj R. Sastry. A contextual attribute–based access control model. *On the Move to Meaningful Internet Systems 2006 OTM 2006 Workshops*, pages 1996–2006, 2006. URL http://www.springerlink.com/index/P63370N24TG16085.pdf. 27, 28

Michael J. Covington, Wende Long, Srividhya Srinivasan, Anind K. Dev, Mustaque Ahamad, and Gregory D. Abowd. Securing context–aware applications using environment roles. In *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 10–20, New York, NY, USA, 2001. ACM. ISBN 1-58113-350-2. doi: http://doi.acm.org/10.1145/373256.373258. 24, 35

Robert Crook, Darrel Ince, and Bashar Nuseibeh. Modelling access policies using roles in requirements engineering. *Information and Software Technology*, 45(14):979–991, 2003. URL http://oro.open.ac.uk/3619/. 22, 36, 37

Robert Crook, Darrel Ince, and Bashar Nuseibeh. On modelling access policies: Relating roles to their organisational context. *Requirements Engineering, IEEE International Conference on*, 0:157–166, 2005. doi: http://doi.IEEEcomputersociety.org/10.1109/RE.2005.48. xix, 36, 37, 38

Frédéric Cuppens and Alexandre Miège. Modelling Contexts in the Or–BAC Model. In *19th Annual Computer Security Applications Conference (ACSAC '03)*, 2003. ISBN 0-7695-2041-3. 30, 41

Frédéric Cuppens, Nora Cuppens-Boulahia, and Meriam Ben Ghorbel. High level conflict management strategies in advanced access control models. *Electron. Notes Theor. Comput. Sci.*, 186:3–26, July 2007. ISSN 1571-0661. doi: 10.1016/j.entcs.2007.01.064. URL http://dx.doi.org/10.1016/j.entcs.2007.01.064. xix, 31

Kirsi Helkala Davrondhon Gafurov and Nils Kalstad Svendsen. Security models for electronic medical record, 2005. 22

Patricia Day and Rudolf Klein. *Accountabilities : five public services*. Tavistock, 1987. URL http://www.worldcat.org/oclc/17412398. 66, 84, 85, 88

S. De Haes and W. Van Grembergen. It governance and its mechanisms. volume 1, 2004. 49

Jan L. G. Dietz. Demo: Towards a discipline of organisation engineering. *European Journal of Operational Research*, 128(2):351–363, 2001. 185

John Dobson and David Martin. Enterprise modeling based on responsibility. In *Trust in Technology: A Socio–Technical Perspective*, volume 36, pages 39–67, 2006. ISBN 978-1-60558-129-3. doi: DOI:10.1007/1-4020-4258-2. 85

John Dobson, Simon Lock, and David Martin. Complexities of multi–organisational error management. In *Proceedings of the 2nd Workshop on Complexity in Design and Engineeing*, pages 131–140, 2006. 70

DoD. United states department of defense, trusted computer system evaluation criteria, dod 5200.28–std, 1985. 23

Melvin Dubnick and Kaifeng Yang. The pursuit of accountability: Promise, problems, and prospects. *The State of Public Administration, Donald Menzel and Harvey White, eds., M.E. Sharpe, Forthcoming*, 2010. 66

Melvin J. Dubnick. Situating accountability: Seeking salvation for the core concept of modern governance (manuscript, university of new hampshire), 2007. 66, 67, 68, 88, 90

Ananda Edirisuriya. Design support for e–commerce information systems using goal, business and process modelling, stockholm university, ph.d. thesis, 2009. 13

Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, and Johannes Willkomm. A method for engineering a true service–oriented architecture. In José Cordeiro and Joaquim Filipe, editors, *ICEIS (3–2)*, pages 272–281, 2008. ISBN 978-989-8111-38-8. URL http://dblp.uni-trier.de/db/conf/iceis/iceis2008-3-2.html#EngelsHHJLRVW08. 81

P. Epstein and R. Sandhu. Engineering of role/permission assignments. In *ACSAC '01: Proceedings of the 17th Annual Computer Security Applications Conference*, page 127, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1405-7. 36

Pete A. Epstein. *Engineering of role/permission assignments*. PhD thesis, Fairfax, VA, USA, 2002. Director–Sandhu, Ravi. 36, 37

Tero Erkkilä. Governance and accountability: A shift in conceptualisation. *Public Administration Quarterly*, 8(3):1–38, 2007. 66, 68, 84, 85, 88

Yanfang Fan, Zhen Han, Jiqiang Liu, and Yong Zhao. A mandatory access control model with enhanced flexibility. In *Proceedings of the 2009 International Conference on Multimedia Information Networking and Security*, volume 01 of *MINES '09*, pages 120–124, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3843-3. doi: 10.1109/MINES.2009.267. URL http://dx.doi.org/10.1109/MINES.2009.267. 23

Christophe Feltus and André Rifaut. An ontology for requirements analysis of managers' policies in financial institutions. In Ricardo Jardim-Gonçalves, Jörg P. Müller, Kai Mertins, and Martin Zelm, editors, *IESA*, pages 27–38. Springer, 2007. ISBN 978-1-84628-857-9. 58

E. B. Fernandez and J. C. Hawkins. Determining role rights from use cases. In *RBAC '97: Proceedings of the second ACM workshop on Role–based access control*, pages 121–125, New York, NY, USA, 1997. ACM. ISBN 0-89791-985-8. doi: http://doi.acm.org/10.1145/266741.266767. 38

David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed nist standard for role–based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001. ISSN 1094-9224. doi: 10.1145/501978.501980. URL http://portal.acm.org/citation.cfm?id=501980. xix, 2, 5, 25, 26, 27, 89, 147

Rodolfo Ferrini and Elisa Bertino. Supporting rbac with xacml+owl. In *SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 145–154, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-537-6. doi: http://doi.acm.org/10.1145/1542207.1542231. 92

T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham. Rowlbac: representing role based access control in owl. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 73–82, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-129-3. doi: http://doi.acm.org/10.1145/1377836.1377849. 92

Tom Fitzpatrick. Horizontal management: Trends in governance and accountability. treasury board of canada secretariat for canadian centre for management development's roundtable on the management of horizontal issues, p. 6., 2006. URL `www.myschool-monecole.gc.ca/research/publications/pdfs/Horiz-Trends-rev.pdf`. 85

Jonathan A. Fox. The uncertain relationship between transparency and accountability. (410729), Aug 2007. URL `http://ideas.repec.org/p/cdl/glinre/410729.html`. 68, 88, 90, 91

Ludwig Fuchs, Günther Pernul, and Ravi S. Sandhu. Roles in information security – a survey and classification of the research area. *Computers & Security*, 30(8):748–769, 2011. 22, 36, 89

Khaled Gaaloul and François Charoy. Task delegation based access control models for workflow systems. In Claude Godart, Norbert Gronau, Sushil K. Sharma, and Gérôme Canals, editors, *I3E*, volume 305 of *IFIP*, pages 400–414. Springer, 2009. ISBN 978-3-642-04279-9. 29

Khaled Gaaloul and H.A. (Erik) Proper. An access control model for organisational management in enterprise architecture. In *Proceedings of the 9th International Conference on Semantics, Knowledge and Grids*. IEEE, 2013. 148

Ashish Garg, Jeffrey Curtis, and Hilary Halper. The financial impact of it security breaches: What do investors think? pages 22–33, 2003. 1

Xiaocheng Ge, Fiona Polack, and Régine Laleau. Secure databases: an analysis of clark–wilson model in a database environment. In *Advanced Information Systems Engineering – 16th International Conference, CAiSE 2004*, pages 7–11, 2004. 25

Christos K. Georgiadis, Ioannis Mavridis, George Pangalos, and Roshan K. Thomas. Flexible team–based access control using contexts. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, SACMAT '01, pages 21–27, New York, NY, USA, 2001. ACM. ISBN 1-58113-350-2. doi: 10.1145/373256.373259. URL `http://doi.acm.org/10.1145/373256.373259`. 31

T. Grandon Gill and Richard C. Hicks. Task complexity and informing science: A synthesis. *Informing Science: The International Journal of an Emerging Transdiscipline*, 9:1–30, 2006. URL `http://www.inform.nu/Articles`. 75

Larry Goldberg, Bettijoyce Lide, Svetlana Lowry, Holly A Massett, Trisha O'Connell, Jennifer Preece, Whitney Quesenbery, and Ben Shneiderman. Usability and accessibility in consumer health informatics current trends and future challenges. *American Journal of Preventive Medicine*, 40(5 Suppl 2):S187–97, 2011. URL `http://www.ncbi.nlm.nih.gov/pubmed/21521594`. 1

K. E. Goodpaster and J. B. Matthews. Can a corporation have a moral conscience? pages 132–141, 1982. 85

J. Richard Hackman. Toward understanding the role of tasks in behavioral research. *Acta Psychologica*, 31:97–128, 1969. 76

Steven De Haes and Wim Van Grembergen. Analysing the relationship between it governance and business/it alignment maturity. In *HICSS*, page 428. IEEE Computer Society, 2008. 49

Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in operating systems. *Commun. ACM*, 19(8):461–471, August 1976. ISSN 0001-0782. doi: 10.1145/360303.360333. URL http://doi.acm.org/10.1145/360303.360333. 24

John C. Henderson and N. Venkatraman. Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1):4–16, 1993. xix, 3, 4, 10, 48, 49

A. R. Hevner, S. T. March, and J. Park. Design science in information systems research. *MIS Quarterly: Management Information Systems*, 28(1):75–105, 2004. 12, 180

Rose Hightower. *Internal controls policies and procedures*. John Wiley and Sons, [S.l.], 2008. ISBN 0-470-28717-9. The book can be consulted by contacting: FP-DI-TPS: Ashford, Robin. 77, 225

Adrian Holzer and Jan Ondrus. Trends in mobile application development. In *MOBILWARE Workshops*, pages 55–64, 2009. 2

Vincent C. Hu and Karen Scarfone. Guidelines for access control system evaluation metrics, September 2012. NISTIR 7874. 24

Ursula Huws, Simone Dahlmann, and Jörg Flecker. Status report on outsourcing of ict and related services in the eu, european foundation for the improvement of living and working conditions, dublin, 2004. 1

Maria-Eugenia Iacob, Dick A. C. Quartel, and Henk Jonkers. Capturing business strategy and value in enterprise architecture to support portfolio valuation. In Chi-Hung Chi, Dragan Gasevic, and Willem-Jan van den Heuvel, editors, *EDOC*, pages 11–20. IEEE, 2012. ISBN 978-1-4673-2444-1. URL http://dblp.uni-trier.de/db/conf/edoc/edoc2012.html#IacobQJ12. 124

IFC. 2002. International Finance Corporation, World Bank Group, Report of the Committee on the Financial Aspects of Corporate Governance (the UK Cadbury Code), London. 46, 61

ISO15504. 2004. ISO/IEC 15504:2004, Software engineering – Process assessment, (parts 1–5), 2003–2006. 49

ISO27000. 2012. ISO/IEC 27000:2012 family of standards – Information Security Management System (incuded are. ISO/IEC 27000, 27002, 27006, 27005, 27004, 27003). xix, 10, 50, 54, 55, 60, 61

ISO38500. 2008. ISO/IEC 38500:2008, International Standard for Corporate Governance of IT. xix, 3, 47, 48, 49, 50, 53, 60, 61

BIBLIOGRAPHY

ISO9000. 2000. ISO/IEC 9000:2000, Quality management. 49

ISO9126-1. 2001. ISO/IEC 9126–1, Information Technology –Software Product Quality– Part 1: Quality Model, International Organization for Standardization, 2001. 16

IT Governance Institute. 2003. IT Governance Institute, Board Briefing on IT Governance, 2nd Edition, 2003. 47

IT Governance Institute, editor. *CobiT 4.1: Framework, Control Objectives, Management Guidelines, Maturity Models*. IT Governance Institute, Rolling Meadows, 2007. xix, 2, 10, 47, 48, 50, 60, 61, 62, 77, 85, 96

IT Governance Institute, editor. *COBIT 5: A Business Framework for the Governance and Management of Enterprise IT*. IT Governance Institute, 2012. URL http://www.isaca.org/COBIT/Pages/default.aspx. 50

ITIL. 2001. IT Infrastructure Library – Service Delivery, The Stationery Office Edition. 49, 50, 124

Sushil Jajodia, Pierangela Samarati, Maria Luisa Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Trans. Database Syst.*, 26(2):214–260, June 2001. ISSN 0362-5915. doi: 10.1145/383891.383894. URL http://doi.acm.org/10.1145/383891.383894. 27

Xin Jin, Ram Krishnan, and Ravi Sandhu. A unified attribute–based access control model covering dac, mac and rbac. In *Proceedings of the 26th Annual IFIP WG 11.3 conference on Data and Applications Security and Privacy*, DBSec'12, pages 41–55, Berlin, Heidelberg, 2012. Springer–Verlag. ISBN 978-3-642-31539-8. doi: 10.1007/978-3-642-31540-4_4. URL http://dx.doi.org/10.1007/978-3-642-31540-4_4. 28

Henk Jonkers, Marc Lankhorst, René Van Buuren, Marcello Bonsangue, and Leendert Van Der Torre. Concepts for modeling enterprise architectures. *International Journal of Cooperative Information Systems*, 13:257–287, 2004. 105, 106

Henk Jonkers, Iver Band, and Dick Quartel. The archisurance case study. *White paper, The Open Group (Spring 2012)*, 2012. 120

James B. D. Joshi. *A generalized temporal role based access control model for developing secure systems*. PhD thesis, West Lafayette, IN, USA, 2003. AAI3113822. 30

Alan H. Karp, Harry Haury, and Michael H. Davis. From abac to zbac : The evolution of access control models from abac to zbac : The evolution of access control models. *Control*, 2009. URL http://www.hpl.hp.com/techreports/2009/HPL-2009-30.html?mtxs=rss-hpl-tr. 27

P. Katranuschkov, A. Gehre, and R. J. Scherer. Reusable process patterns for collaborative work environments in aec. In *Proceedings of the 13th International Conference on Concurrent Enterprising*, 2007. 79

T. Kendrick. *Results Without Authority: Controlling a Project When the Team Doesn't Report to You*. AMACOM, 2012. ISBN 9780814417812. URL http://books.google.be/books?id=QeiLltM5mKsC. 225

Axel Kern, Martin Kuhlmann, Andreas Schaad, and Jonathan Moffett. Observations on the role life–cycle in the context of enterprise security management. In *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies*, SACMAT '02, pages 43–51, New York, NY, USA, 2002. ACM. ISBN 1-58113-496-7. doi: 10.1145/507711.507718. URL http://doi.acm.org/10.1145/507711.507718. 35

Dae-Kyoo Kim, Indrakshi Ray, Robert B. France, and Na Li. Modeling role–based access control using parameterized uml models. In Michel Wermelinger and Tiziana Margaria, editors, *FASE*, volume 2984 of *Lecture Notes in Computer Science*, pages 180–193. Springer, 2004. ISBN 3-540-21305-8. URL http://dblp.uni-trier.de/db/conf/fase/fase2004.html#KimRFL04. 152

Thomas Kreifelts, Elke Hinrichs, and Gerd Woetzel. Sharing to–do lists with a distributed task manager. In *Proceedings of the Third European Conference on Computer Supported Cooperative Work (ECSCW–93*, pages 31–46. Kluwer Academic Publishers, 1993. 86

D. Richard Kuhn, Edward J. Coyne, and Timothy R. Weil. Adding attributes to role–based access control. *Computer*, 43(6):79–81, 2010. URL http://csrc.nist.gov/groups/SNS/rbac/. 27, 28

Butler W. Lampson. Protection. *SIGOPS Oper. Syst. Rev.*, 8(1):18–24, January 1974. ISSN 0163-5980. doi: 10.1145/775265.775268. URL http://doi.acm.org/10.1145/775265.775268. 22

Bo Lang, Ian Foster, Frank Siebenlist, Rachana Ananthakrishnan, and Tim Freeman. A flexible attribute based access control method for grid computing. *Journal of Grid Computing*, 7(2): 169–180, 2008. URL http://www.springerlink.com/index/10.1007/s10723-008-9112-1. 27, 28

M. Lankhorst. Archimate language primer, 2004. 5, 81, 104

Marc Lankhorst and Hans van Drunen. Enterprise Architecture Development and Modelling. Combining TOGAF and ArchiMate. pages 1–16, March 2007. 4

C. Kenneth Laudon and P. Jane Laudon. Essentials of management information systems: Organization & technology in the networked enterprise. 2001. 84, 85, 87

Richard Lenz and Klaus Kuhn. A strategic approach for business–it alignment in health information systems. *On The Move to Meaningful Internet Systems 2003 CoopIS DOA and ODBASE*, pages 178–195, 2003. URL http://www.springerlink.com/content/eldbany5kwy0nhmf. 3

Peter Liebwein. Risk models for capital adequacy: Applications in the context of solvency ii and beyond. *The Geneva Papers on Risk and Insurance – Issues and Practice*, 31(3):528–550, 2006. URL http://EconPapers.repec.org/RePEc:pal:gpprii:v:31:y:2006:i:3:p:528-550. 4

Ann Lindsay, Denise Downs, and Ken Lunn. Business processes – attempts to find a definition. *Information and Software Technology*, 45(15):1015 – 1019, 2003. ISSN 0950-5849. doi: 10.1016/S0950-5849(03)00129-0. URL http://www.sciencedirect.com/science/article/pii/S0950584903001290. Special Issue on Modelling Organisational Processes. 50

Steven B. Lipner. Non–discretionary controls for commercial applications. In *IEEE Symp. on Security and Privacy*, pages 2–10, Oakland, CA, 1982. 23

BIBLIOGRAPHY

Jerry N. Luftman. *Competing in the Information Age: Strategic Alignment in Practice*. Oxford University Press, Inc., New York, NY, USA, 1st edition, 1996. ISBN 0195090160. 48

Fred C. Lunenburg. Organizational structure: Mintzbergs framework. *International Journal of Scholarly, Academic, Intellectual Diversity*, 14(1), 2012. 10

L. de Las Casas M. Blagescu and R. Lloyd. Pathways to accountability: The global accountability framework. *One World Trust*, 2011. 67

Rik Maes, Daan Rijsenbrij, Onno Truijens, and Hans Goedvolk. Redefining business – IT alignment through a unified framework. In *Proceedings of the Landelijk Architectuur Congres*, Amsterdam, 2000. URL http://home.hetnet.nl/~{}daan.rijsenbrij/uvacap/alignment/index.html. 48

Paul Maglio and Jim Spohrer. Fundamentals of service science. *Journal of the Academy of Marketing Science*, 36(1):18–20, 2008. ISSN 0092-0703. doi: 10.1007/s11747-007-0058-9. URL http://dx.doi.org/10.1007/s11747-007-0058-9. 184

Salvatore T. March and Gerald F. Smith. Design and natural science research on information technology. *Decis. Support Syst.*, 15(4):251–266, 1995. ISSN 0167-9236. doi: http://dx.doi.org/10.1016/0167-9236(94)00041-2. 13, 16

Henk Jonkers Maria-Eugenia Iacob and Martijn Wiering. Towards a uml profile for the archimate language, 2004. 105, 106

David Martin, Mark Rouncefield, Jacki O'Neill, Mark Hartswood, and Dave Randall. Timing in the art of integration: 'that's how the bastille got stormed'. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, GROUP '05, pages 313–322, New York, NY, USA, 2005. ACM. ISBN 1-59593-223-2. doi: http://doi.acm.org/10.1145/1099203.1099256. URL http://doi.acm.org/10.1145/1099203.1099256. 68, 84, 87

Fabio Massacci. Reasoning about security: A logic and a decision method for role–based access control. In Dov M. Gabbay, Rudolf Kruse, Andreas Nonnengart, and Hans Jürgen Ohlbach, editors, *Qualitative and Quantitative Practical Reasoning, First International Joint Conference on Qualitative and Quantitative Practical Reasoning ECSQARU–FAPR 97, Bad Honnef, Germany, June 9–12, 1997, Proceedings*, volume 1244 of *Lecture Notes in Computer Science*, pages 421–435. Springer, 1997. ISBN 3-540-63095-3. 27

John Mclean. Reasoning about security models. *Security and Privacy, IEEE Symposium on*, 0: 123, 1987. ISSN 1540-7993. doi: http://doi.ieeecomputersociety.org/10.1109/SP.1987.10020. 23

Nikola Milanovic, Mario Cartsburg, Ralf Kutsche, Jürgen Widiker, and Frank Kschonsak. Model–based interoperability of heterogeneous information systems: An industrial case study. In Richard Paige, Alan Hartman, and Arend Rensink, editors, *Model Driven Architecture – Foundations and Applications*, volume 5562 of *Lecture Notes in Computer Science*, pages 325–336. Springer Berlin, Heidelberg, 2009. ISBN 978-3-642-02673-7. URL http://dx.doi.org/10.1007/978-3-642-02674-4_24. 1

Henri Mintzberg. Structure in fives: Designing effective organisations. In *Prentice Hall*, 1992. 37

MIT. 2002. Massachusetts Institute of Technology, Center for Information Systems, Research Working Paper No. 326; April 2002. 3, 47, 61

Jonathan D. Moffett. Network and distributed systems management. chapter Specification of management policies and discretionary access control, pages 455–480. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994. ISBN 0-201-62745-0. URL http://dl.acm.org/citation.cfm?id=184430.184484. 24

Richard Mulgan. Accountability: An ever–expanding concept? *Public Administration*, 78(3): 555–573, 2000. URL http://dspace.anu.edu.au/handle/1885/41945. 66, 67, 68, 84, 85, 87, 88, 90

Gustaf Neumann and Mark Strembeck. A scenario–driven role engineering process for functional rbac roles. In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 33–42, New York, NY, USA, 2002. ACM. ISBN 1-58113-496-7. doi: http://doi.acm.org/10.1145/507711.507717. xix, 39, 40, 41

Object Management Group (OMG). Uml 2.4.1 superstructure specification, 2011. 152

Alan C. O'Connor and Ross J. Loomi. Economic benefits of role based access control analyzes economic value of rbac for the enterprise and for the national economy, and provides quantitative economic benefits of rbac per employee for adopting firms. 2011. xix, 2, 3, 21, 28

OECD. 2004. Organisation for Economic Co–operation and Development. OECD Principles of Corporate Governance: 2004. OECD, 2004. 46, 61

Sejong Oh and Seog Park. Task–role–based access control model. *Inf. Syst.*, 28(6):533–562, September 2003. ISSN 0306-4379. doi: 10.1016/S0306-4379(02)00029-7. URL http://dx.doi.org/10.1016/S0306-4379(02)00029-7. 29

Charles A. O'Reilly and Jennifer Chatman. Organizational commitment and psychological attachment: The effects of compliance, identification, and internalization on prosocial behavior. *Journal of Applied Psychology*, 71(3):492–499, 1986. doi: 10.1037/0021-9010.71.3.492. URL http://dx.doi.org/10.1037/0021-9010.71.3.492. 68

Jack A. Orenstein. Supporting retrievals and updates in an object/relational mapping system. *IEEE Data Eng. Bull.*, 22(1):50–54, 1999. 81

Alexander Osterwalder. *The Business Model Ontology: a proposition in a design science approach*. Dissertation, Université de Lausanne, Ecole des Hautes Etudes Commerciales, 2004. URL http://www.hec.unil.ch/aosterwa/PhD/. 13

Prashant C. Palvia, En Mao, Khalid S. Soliman, and A. F. Salam. Management information systems research: What's there in a methodology? *Communications of the Association for Information Systems*, 11(1):289–309, 2003. 15, 16

Christine Parent and Stefano Spaccapietra. Database integration: The key to data interoperability. In *Advances in Object–Oriented Data Modeling*, pages 221–253. 2000. URL http://dblp.uni-trier.de/db/books/collections/Papazoglou2000.html#ParentS00. 16, 113, 114, 182

Jaehong Park and Ravi Sandhu. Towards usage control models: beyond traditional access control. In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 57–64, New York, NY, USA, 2002. ACM. ISBN 1-58113-496-7. doi: http://doi.acm.org/10.1145/507711.507722. xix, 31, 32

Jaehong Park and Ravi Sandhu. The uconabc usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, 2004. ISSN 1094-9224. doi: http://doi.acm.org/10.1145/984334.984339. xix, 32

Hugh Parkes. It governance and outsourcing. *Information Systems Control Journal*, 5, 2004. 47

Fabio Paterno. Task models in interactive software systems. In S. K. Chang, editor, *Handbook of Software Engineering and Knowledge*. World Scientific Publishing Co, 2001. 75, 79

Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *J. of Management Information Systems*, 24(3):45–77, 2008. 14

Joe Peppard. Beyond strategic information systems: towards an is capability. *The Journal of Strategic Information Systems*, 13(2):167–194, 2004. 1

Michaël Petit. Some methodological clues for defining a unified enterprise modelling language. In *Proceedings of the IFIP TC5/WG5.12 International Conference on Enterprise Integration and Modeling Technique: Enterprise Inter– and Intra–Organizational Integration: Building International Consensus*, ICEIMT '01, pages 359–369, Deventer, The Netherlands, The Netherlands, 2003. Kluwer, B.V. ISBN 1-4020-7277-5. URL http://dl.acm.org/citation.cfm?id=647037.715576. 182, 236, 239, 240, 242, 258

C. J. Prendergast. A theory of responsibility in organizations. volume 13(3), pages 387–400, 1995. 64, 84, 86, 87

T. Priebe, W. Dobmeier, and N. Kamprath. *Supporting Attribute–based Access Control with Ontologies*, pages 465–472. IEEE, 2006. URL http://IEEExplore.IEEE.org/lpdocs/epic03/wrapper.htm?arnumber=1625344. 28

Torsten Priebe, Wolfgang Dobmeier, Christian Schläger, and Nora Kamprath. Supporting attribute–based access control in authorization and authentication infrastructures with ontologies. *JSW*, 2(1):27–38, 2007. xix, 28

Brigid Proctor. Supervision–competence, confidence, accountability. *British Journal of Guidance and Counselling*, 22(3):309–318, 1994. doi: 10.1080/03069889408253676. 77

Indrakshi Ray and Manachai Toahchoodee. A spatio–temporal role–based access control model. In *Proceedings of the 21st annual IFIP WG 11.3 working conference on Data and applications security*, pages 211–226, Berlin, Heidelberg, 2007. Springer–Verlag. ISBN 978-3-540-73533-5. URL http://dl.acm.org/citation.cfm?id=1770560.1770582. 30

Indrakshi Ray, Na Li, Robert France, and Dae-Kyoo Kim. Using uml to visualize role–based access control constraints. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, SACMAT '04, pages 115–124, New York, NY, USA, 2004. ACM. ISBN 1-58113-872-5. doi: 10.1145/990036.990054. URL http://doi.acm.org/10.1145/990036.990054. 152, 154

Blaize Horner Reich and Izak Benbasat. Measuring the linkage between business and information technology objectives. *MIS Quarterly*, 20(1):55–81, 1996. ISSN 0276-7783. 48

ISO Interim Report. 2007. Interim Report, Study Group on ICT Governance, ISO/IEC JTC1/SC7 /N3861. 49

Gail Ridley, Judy Young, and Peter Carroll. Cobit and its utilization: A framework from the literature. In *HICSS*, 2004. URL http://dblp.uni-trier.de/db/conf/hicss/hicss2004-8.html#RidleyYC04. 49

Haio Roeckle, Gerhard Schimpf, and Rupert Weidinger. Process–oriented approach for role–finding to implement role–based security administration in a large industrial organization. In *RBAC '00: Proceedings of the fifth ACM workshop on Role–based access control*, pages 103–110, New York, NY, USA, 2000. ACM. ISBN 1-58113-259-X. doi: http://doi.acm.org/10.1145/344287.344308. 36

Barbara S. Romzek and Melvin J. Dubnick. Accountability in the Public Sector: Lessons from the Challenger Tragedy. *Public Administration Review*, 47(3):227–238, 1987. ISSN 00333352. doi: 10.2307/975901. URL http://dx.doi.org/10.2307/975901. 66

Ravi Sandhu and Jaehong Park. Usage control: A vision for next generation access control. In *MMM–ACNS*, pages 17–31, 2003. xix, 32, 33, 34, 35

Ravi S. Sandhu. Separation of duties in computerized information systems. In *DBSec*, pages 179–190, 1990. 92

Maung K. Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. Action design research. *MIS Q.*, 35(1):37–56, March 2011. ISSN 0276-7783. URL http://dl.acm.org/citation.cfm?id=2017483.2017487. xix, 14

Michael E. Shin and Gail-Joon Ahn. Uml–based representation of role–based access control. In *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, WETICE '00, pages 195–200, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0798-0. URL http://dl.acm.org/citation.cfm?id=647068.715641. 152

A. Sinclair. The chameleon of accountability: Forms and discourses. *Accounting, Organizations and Society*, 20(2–3):219–237, February 1995. ISSN 03613682. doi: 10.1016/0361-3682(93)E0003-Y. URL http://dx.doi.org/10.1016/0361-3682(93)E0003-Y. 66, 68, 84, 85, 88

Dirk Sliwka. On the notion of responsibility in organizations. volume 22(2), 2006. 64, 84, 86

Ian Sommerville. Models for responsibility assignment. In *Responsibility and Dependable Systems*. Springer, 2007a. 69, 70, 84, 86, 87, 88, 90, 91, 95

Ian Sommerville. Causal responsibility model. In *Responsibility and Dependable Systems*. Springer, 2007b. 70, 87

Ian Sommerville, Russell Lock, Tim Storer, and John Dobson. Deriving information requirements from responsibility models. In *CAiSE '09: Proceedings of the 21st International Conference on Advanced Information Systems Engineering*, pages 515–529, Berlin, Heidelberg, 2009a. Springer–Verlag. ISBN 978-3-642-02143-5. doi: http://dx.doi.org/10.1007/978-3-642-02144-2_40. 70, 71, 85

BIBLIOGRAPHY

Ian Sommerville, Tim Storer, and Russell Lock. Responsibility modelling for civil emergency planning, 2009b. URL http://www.ncjrs.gov/App/Publications/abstract.aspx?ID=250623. 68, 71, 84, 86, 87

Suzanne Soroczak and David W. McDonald. Collaborating over project schedules. In *Proceedings of Supporting the Social Side of Large Scale Software Development – CSCW Workshop '06*, pages 43–46, Banff, AB, 2006. 78

SOX. Sarbanes–oxley act of 2002, united states code, pl 107–204, 116 stat 745. July 2002. 3, 59, 60, 61

Richard A. Spinello. *Case Studies in Information and Computer Ethics*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 1996. ISBN 013533845X. 84, 85, 87

Mario Spremić. Standards and frameworks for information system security auditing and assurance. *Proceedings of the World Congress on Engineering 2011*, 1:251–266, 2011. ISSN 2078-0958. doi: http://dx.doi.org/10.1016/0167-9236(94)00041-2. 1, 49

Mario Spremić and Hrvoje Spremić. Measuring it governance maturity: evidences from using regulation framework in the republic croatia. In *Proceedings of the 5th European conference on European computing conference*, ECC'11, pages 98–104, Stevens Point, Wisconsin, USA, 2011. World Scientific and Engineering Academy and Society (WSEAS). ISBN 978-960-474-297-4. 49

Bernd Carsten Stahl. Accountability and reflective responsibility in information systems. 195: 51–68, 2006. URL http://dx.doi.org/10.1007/0-387-31168-8_4. 68, 84, 85, 86, 87, 90

Michael Stieghahn and Thomas Engel. Law–aware access control: About modeling context and transforming legislation. In Kumiyo Nakakoji, Yohei Murakami, and Eric McCready, editors, *New Frontiers in Artificial Intelligence*, volume 6284 of *Lecture Notes in Computer Science*, pages 73–86. Springer Berlin, Heidelberg, 2010. ISBN 978-3-642-14887-3. URL http://dx.doi.org/10.1007/978-3-642-14888-0_7. 2

Tim Storer and Russell Lock. Modelling responsibility. project working paper 7, indeed project, 2008. 68, 84, 86, 87

Ros Strens and John Dobson. How responsibility modelling leads to security requirements. In *Proceedings on the 1992–1993 workshop on New security paradigms*, NSPW '92–93, pages 143–149, New York, NY, USA, 1993. ACM. ISBN 0-8186-5430-9. doi: http://doi.acm.org/10.1145/283751.283828. URL http://doi.acm.org/10.1145/283751.283828. 69, 84, 86

Felix B. Tan and R. B. Gallupe. Aligning business and information systems thinking: a cognitive approach. *IEEE Transactions on Engineering Management*, 53(2):223–237, 2006. URL http://apps.isiknowledge.com/full_record.do?product=UA&search_mode=Refine&qid=61&SID=U2gn3a7aG7P@43I1e2l&page=2&doc=53&colname=WOS. 3

The Open Group. Togaf 8.1.1 online, part iv: Resource base, architecture governance, 2006. URL http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap26.html. 46, 48, 50

The Open Group. TOGAF 9 – The Open Group Architecture Framework Version 9, 2009. URL http://www.opengroup.org/togaf/. 81

The Open Group. *ArchiMate® 2.0 Specification*. Van Haren Publishing, The Netherlands, 2012. URL http://pubs.opengroup.org/architecture/archimate2-doc/. xx, xxi, xxiii, 81, 89, 104, 107, 108, 110, 149, 229, 230, 231

The World Bank. Managing development: The governance dimension, discussion paper. 1991. 46

Roshan K. Thomas. Team–based access control (tmac): a primitive for applying role–based access controls in collaborative environments. In *Proceedings of the second ACM workshop on Role–based access control*, RBAC '97, pages 13–19, New York, NY, USA, 1997. ACM. ISBN 0-89791-985-8. doi: 10.1145/266741.266748. URL http://doi.acm.org/10.1145/266741.266748. 31

Roshan K. Thomas and Ravi S. Sandhu. Task–based authorization controls (tbac): A family of models for active and enterprise–oriented autorization management. In *Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects*, pages 166–181, London, UK, UK, 1998. Chapman & Hall, Ltd. ISBN 0-412-82090-0. URL http://dl.acm.org/citation.cfm?id=646115.679940. 29

D. Thomsen, R. O'Brien, and C. Payne. Napoleon: network application policy environment. In *RBAC '99: Proceedings of the fourth ACM workshop on Role–based access control*, pages 145–152, New York, NY, USA, 1999. ACM. ISBN 1-58113-180-1. doi: http://doi.acm.org/10.1145/319171.319185. 36

Jaideep Vaidya, Vijayalakshmi Atluri, and Qi Guo. The role mining problem: Finding a minimal descriptive set of roles. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*, SACMAT '07, pages 175—184, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-745-2. doi: 10.1145/1266840.1266870. URL http://doi.acm.org/10.1145/1266840.1266870. 35

Sven van Dijk, Iver Band, Henry Franken, Bas van Gils, and Rob Kroeses. Building an effective enterprise architecture capability – using togaf®and the grid approach. *White paper, BiZZdesign*, 2013. 124

François Vernadat. Enterprise modelling and integration. In *ICEIMT*, pages 25–33, 2002. 93, 96

Marco Vicente, Nelson Gama, and Miguel Mira da Silva. The value of itil in enterprise architecture. In *17th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2013)*. IEEE, IEEE, september 2013. 124

Nicole A. Vincent. A structured taxonomy of responsibility concepts. *Moral Responsibility: Beyond free will and determinism, Nicole A.Vincent, Van de Poel and Van den Hoven*, pages 15–35, August 2011. URL http://ssrn.com/abstract=1662385. xix, 64, 65, 70, 74, 84, 86, 87

Mary J. Waller. Keeping the pins in the air: How work groups juggle multiple tasks. *Interdisciplinary Studies of Work Teams*, 4:217–247, 1997. 75

Alf Inge Wang. xperience paper: Using xml to implement a workflow tool. In *Proceedings of the 3rd Annual IASTED International Conference Software Engineering and Applications*, 1999. 86

Phyl Webb, Carol Pollard, and Gail Ridley. Attempting to define it governance: Wisdom or folly? In *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, page 194.1, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2507-5. doi: http://dx.doi.org/10.1109/HICSS.2006.68. 47, 48, 61

Peter Weill and Jeanne W. Ross. *IT Governance : How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business School Pr., Boston, Mass., 2004. ISBN 1-59139-253-5. 47

Stephen A. White. Business process modeling notation (bpmn) version 1.0. Technical report, BPMI.org, 2004. 75, 76, 79

Roel Wieringa. Design science methodology: Principles and practice. In *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering*, volume 2 of *ICSE '10*, pages 493–494, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-719-6. doi: 10.1145/1810295.1810446. URL http://doi.acm.org/10.1145/1810295.1810446. 16

Eric Yu and Lin Liu. Modelling trust in the i* strategic actors framework. In *Proc. of the 3rd Workshop on Deception, Fraud and Trust in Agent Societies at Agents2000*, pages 3–4, 2000. xix, 38, 39

Eric S. K. Yu. Towards modeling and reasoning support for early–phase requirements engineering. In *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, RE '97, pages 226–, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-8186-7740-6. URL http://dl.acm.org/citation.cfm?id=827255.827807. 77, 79

Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, Toronto, Ont., Canada, Canada, 1996. UMI Order No. GAXNN–02887 (Canadian dissertation). 76, 185

Martin Zelm, François Vernadat, and Kurt Kosanke. The cimosa business modelling process. *Comput. Ind.*, 27(2):123–142, 1995. ISSN 0166-3615. doi: http://dx.doi.org/10.1016/0166-3615(95)00018-2. 95

Xinwen Zhang, Jaehong Park, Francesco Parisi-Presicce, and Ravi Sandhu. A logical specification for usage control. In *SACMAT '04: Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 1–10, New York, NY, USA, 2004. ACM. ISBN 1-58113-872-5. doi: http://doi.acm.org/10.1145/990036.990038. xix, 32, 33

Chen Zhao, Nuermaimaiti Heilili, Shengping Liu, and Zuoquan Lin. Representation and reasoning on rbac: a description logic approach. In *Proceedings of the Second international conference on Theoretical Aspects of Computing*, ICTAC'05, pages 381–393, Berlin, Heidelberg, 2005. Springer–Verlag. ISBN 3-540-29107-5, 978-3-540-29107-7. doi: 10.1007/11560647_25. URL http://dx.doi.org/10.1007/11560647_25. 27

Srdjan Zivkovic, Harald Kühn, and Dimitris Karagiannis. Facilitate modelling using method integration: An approach using mappings and integration rules. In Hubert Österle, Joachim Schelp, and Robert Winter, editors, *ECIS*, pages 2038–2049. University of St. Gallen, 2007. 113, 124, 182

Appendices

Appendix A

List of the responsibilities from the doctors and chief doctors' scenarii

A. LIST OF THE RESPONSIBILITIES FROM THE DOCTORS AND CHIEF DOCTORS' SCENARII

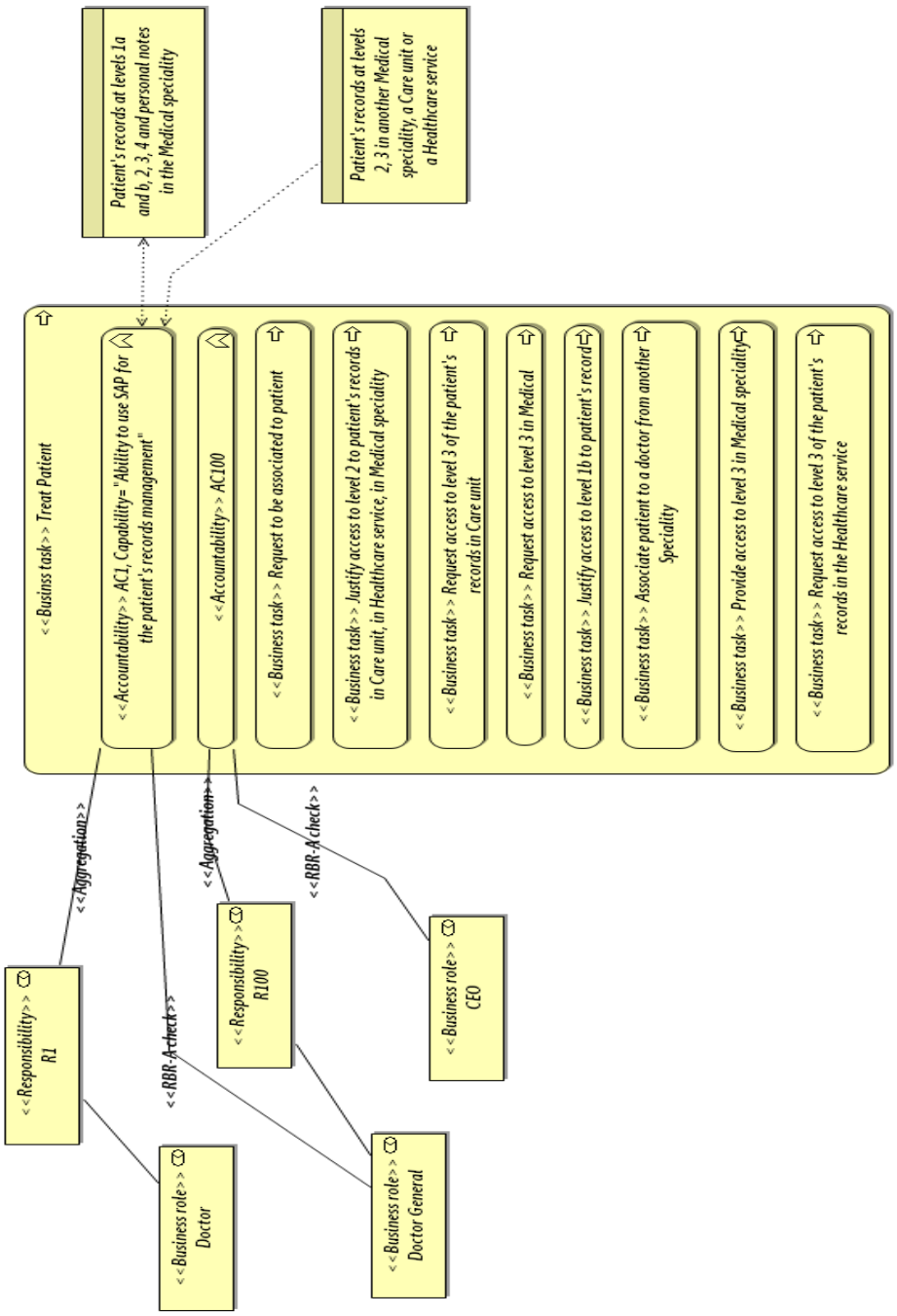


Figure A.1: Responsibilities *R1* and *R100*

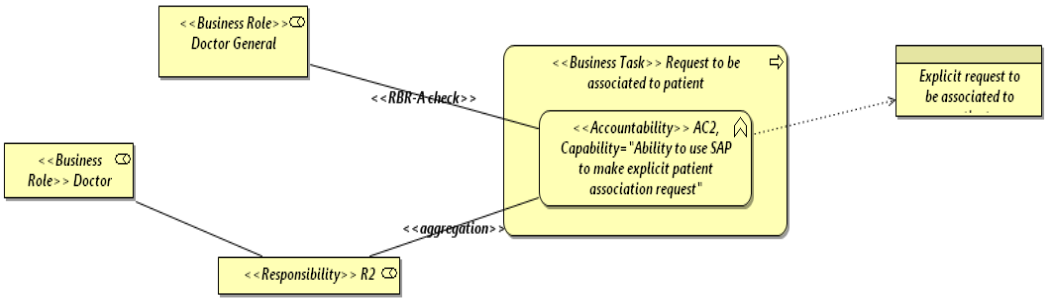


Figure A.2: Responsibility *R2*

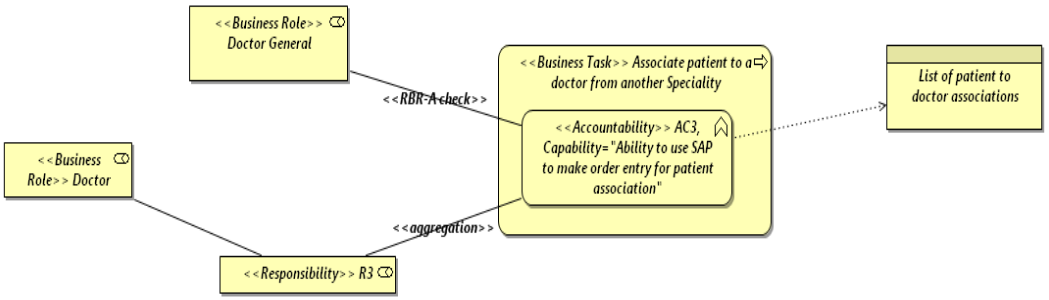


Figure A.3: Responsibility *R3*

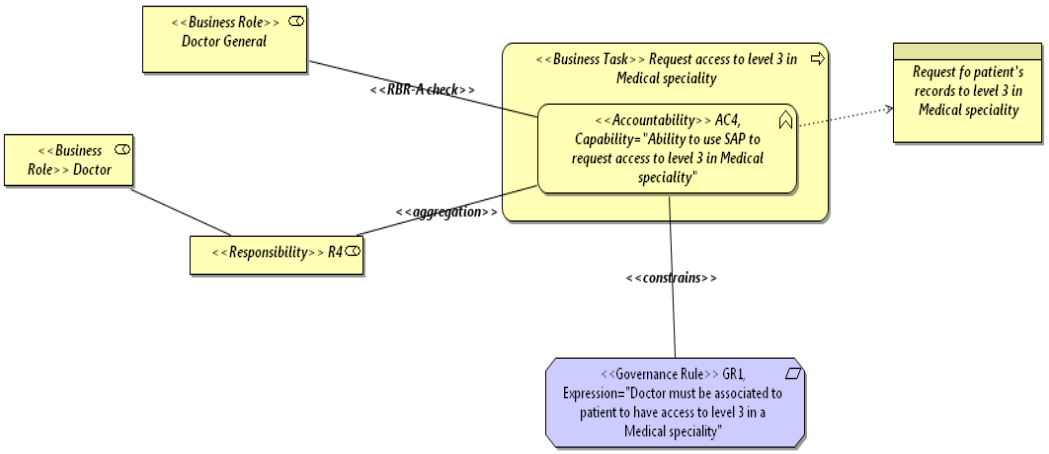


Figure A.4: Responsibility *R4*

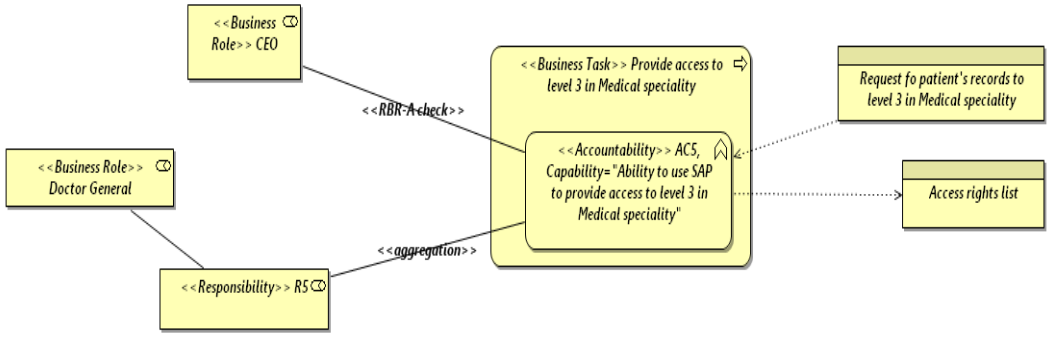


Figure A.5: Responsibility *R5*

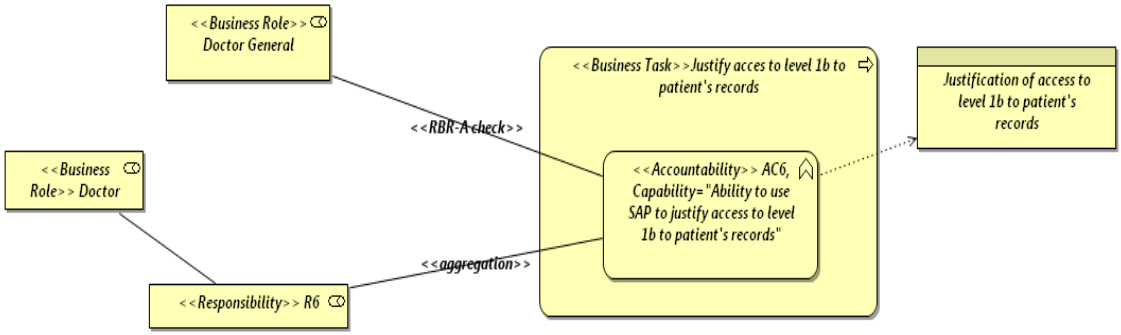


Figure A.6: Responsibility *R6*

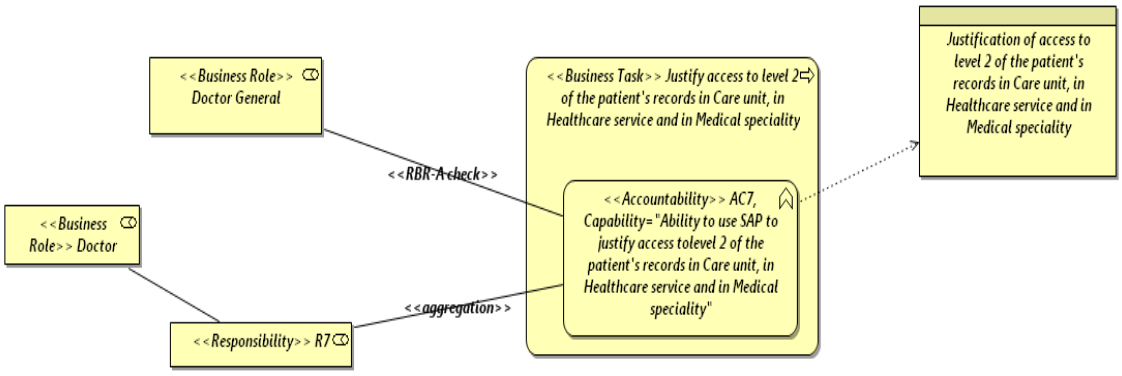


Figure A.7: Responsibility *R7*

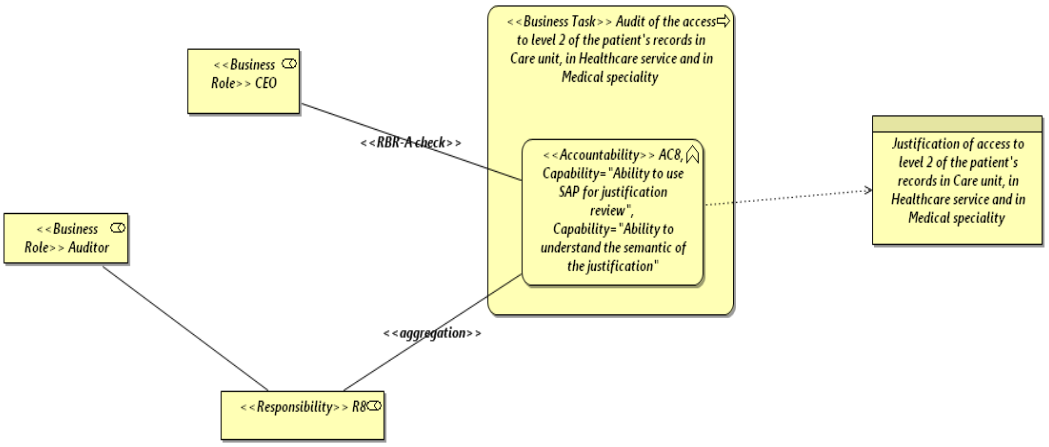


Figure A.8: Responsibility *R8*

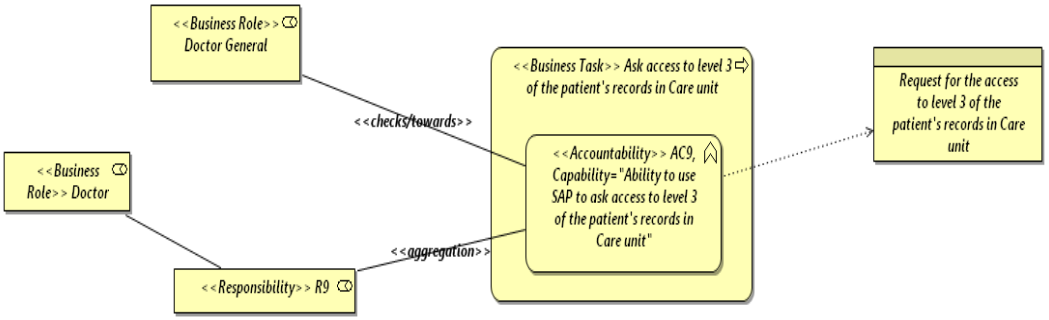


Figure A.9: Responsibility *R9*

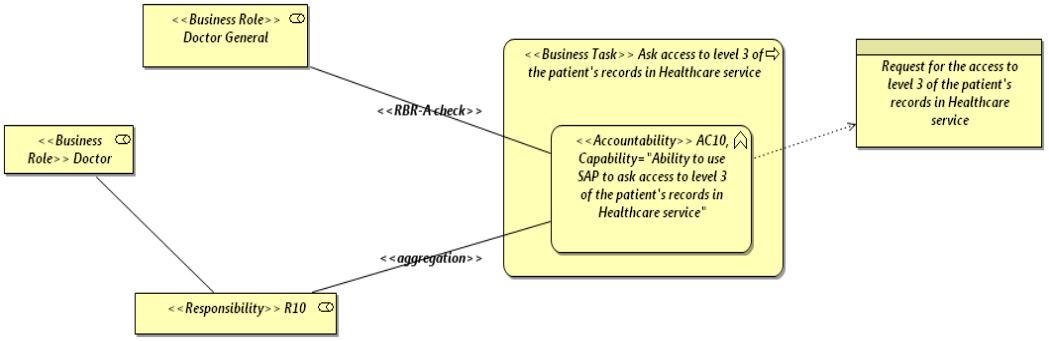


Figure A.10: Responsibility *R10*

A. LIST OF THE RESPONSIBILITIES FROM THE DOCTORS AND CHIEF DOCTORS' SCENARII

Appendix B

List of the responsibilities from the medical secretaries' scenarii

B. LIST OF THE RESPONSIBILITIES FROM THE MEDICAL SECRETARIES' SCENARII

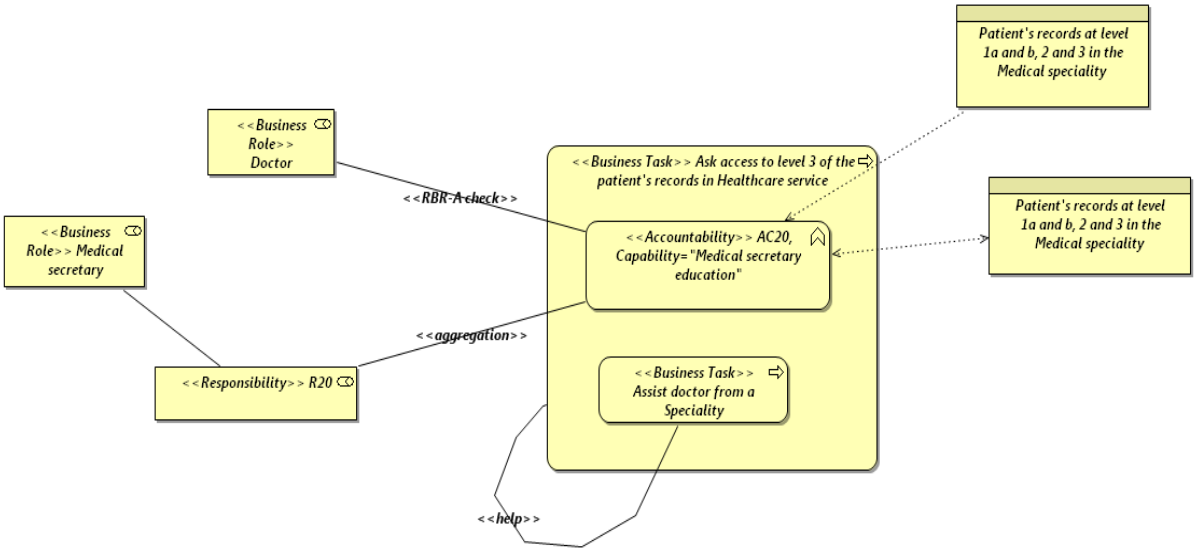


Figure B.1: Responsibility *R20*

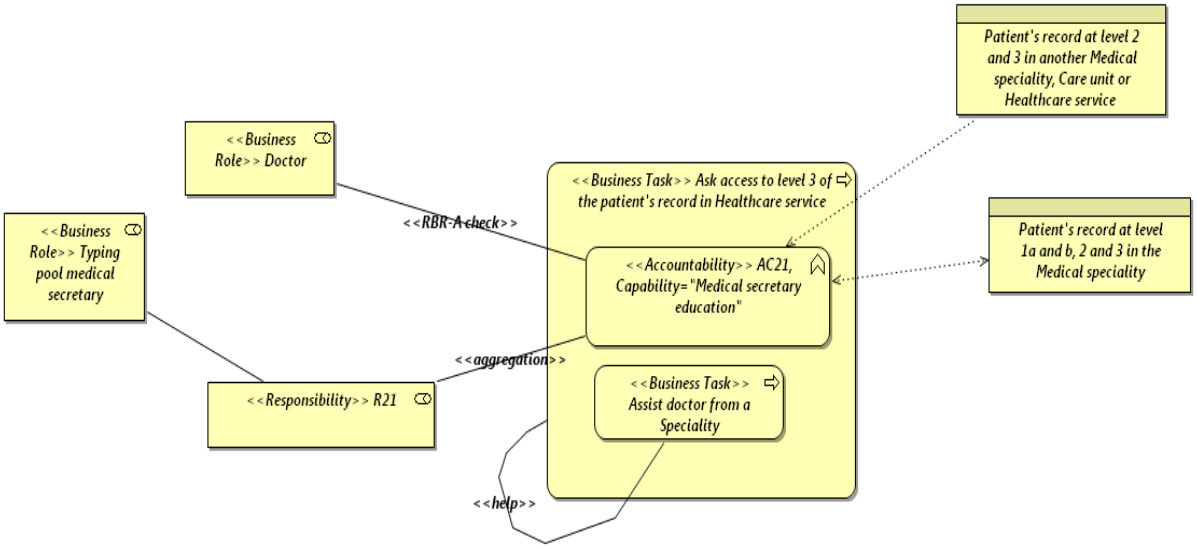


Figure B.2: Responsibility *R21*

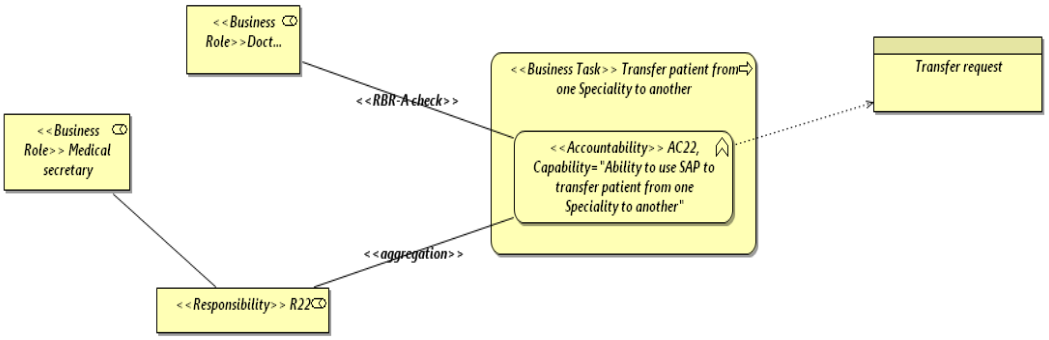


Figure B.3: Responsibility *R22*

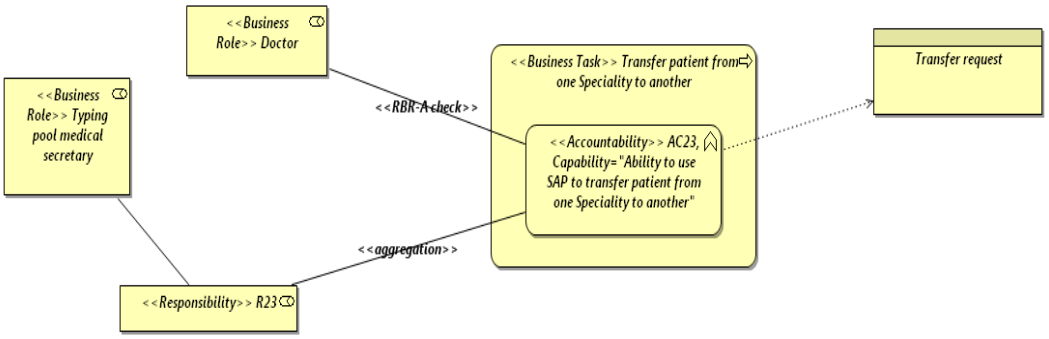


Figure B.4: Responsibility *R23*

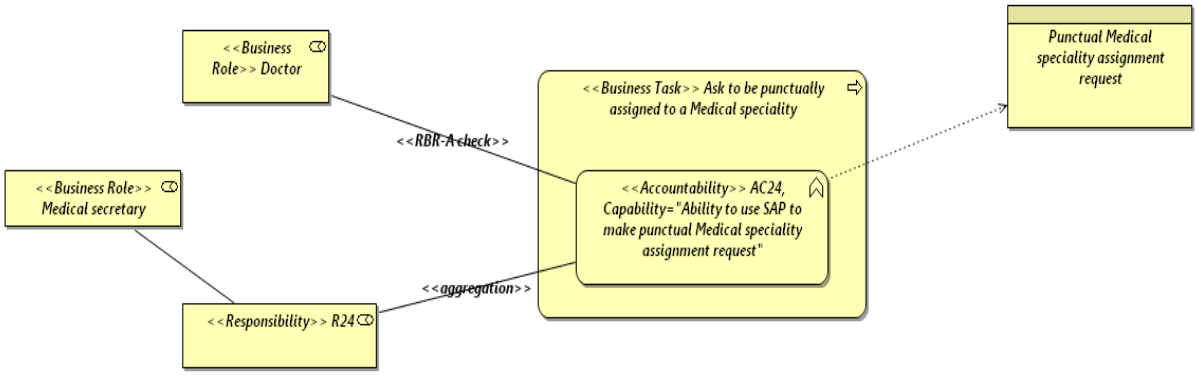


Figure B.5: Responsibility *R24*

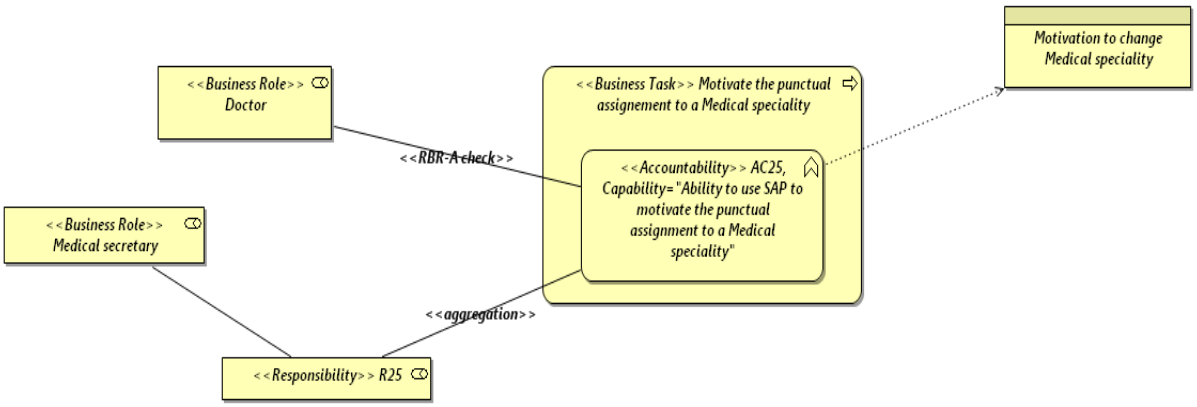


Figure B.6: Responsibility *R25*

Appendix C

List of the responsibilities from the nurses and healthcare specialists' scenarii

C. LIST OF THE RESPONSIBILITIES FROM THE NURSES AND HEALTHCARE SPECIALISTS' SCENARII

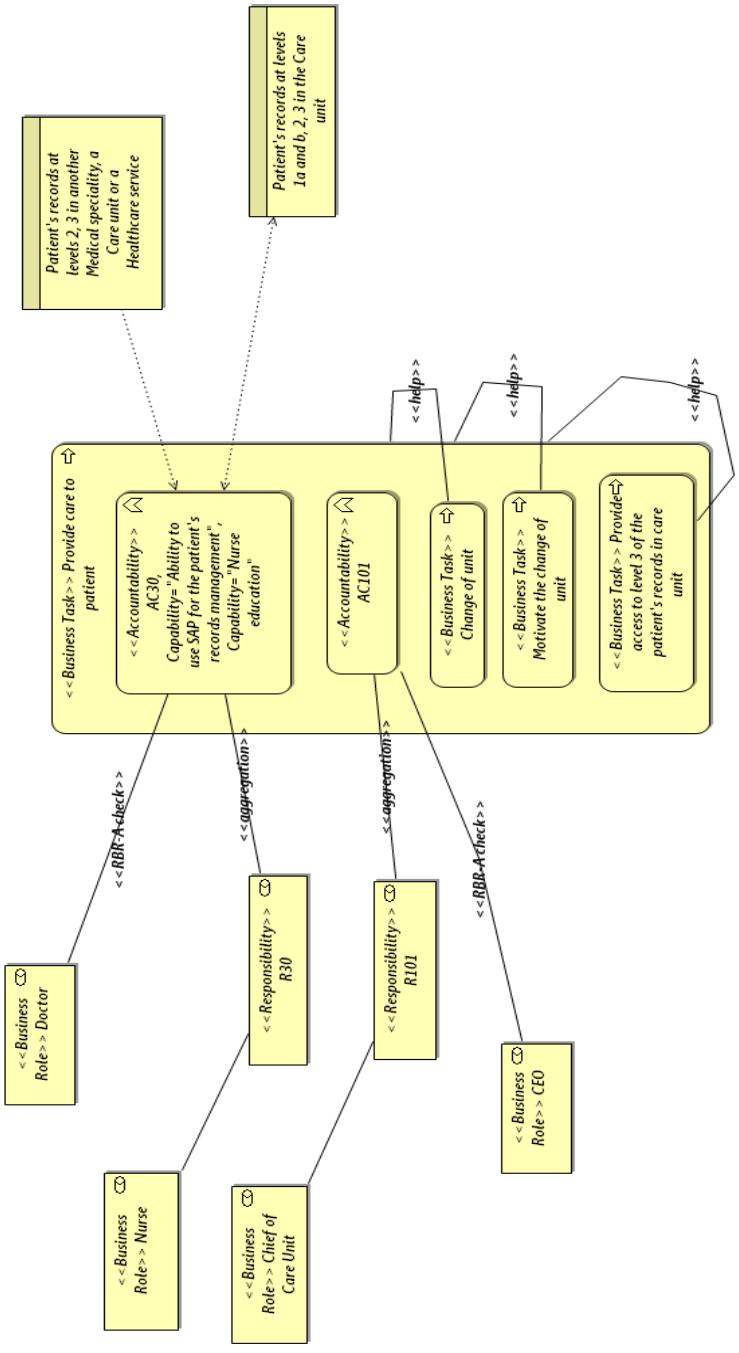


Figure C.1: Responsibilities *R30* and *R101*

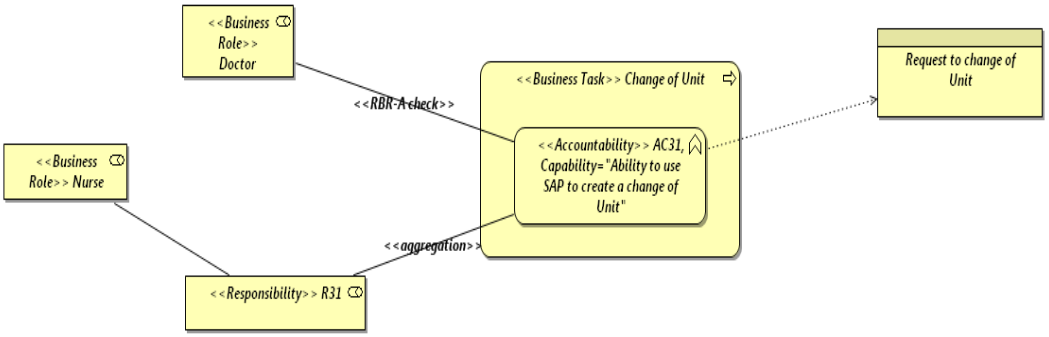


Figure C.2: Responsibility *R31*

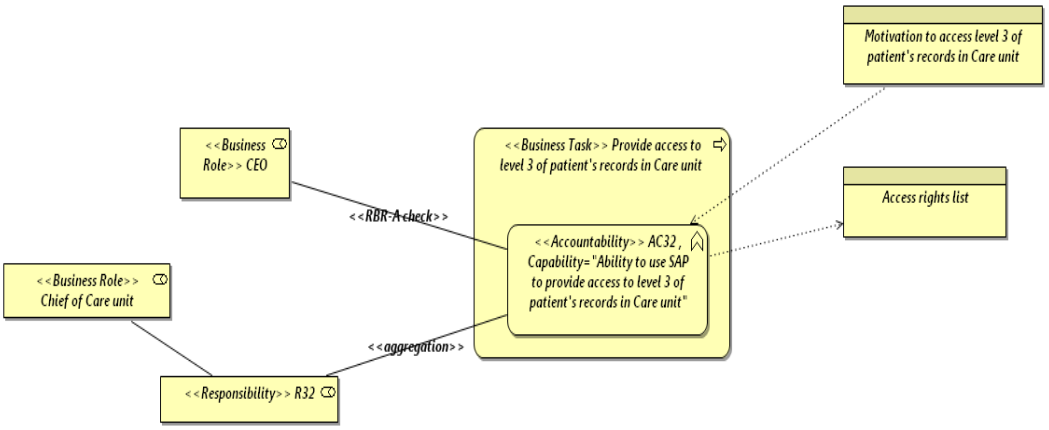


Figure C.3: Responsibility *R32*

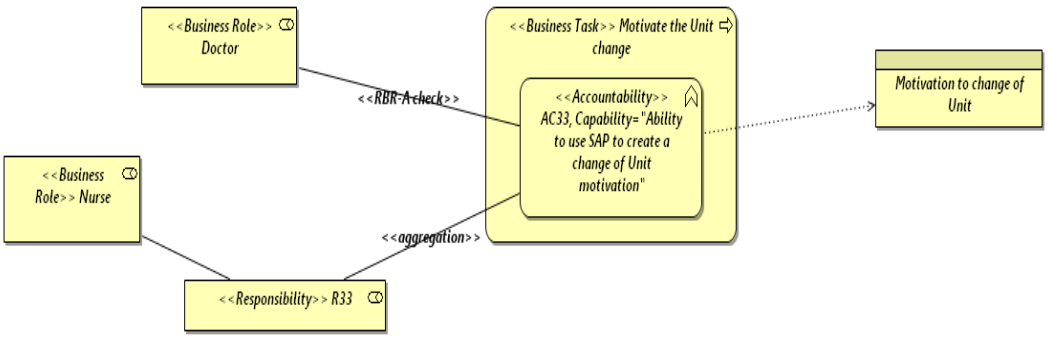


Figure C.4: Responsibility *R33*

C. LIST OF THE RESPONSIBILITIES FROM THE NURSES AND HEALTHCARE SPECIALISTS' SCENARII

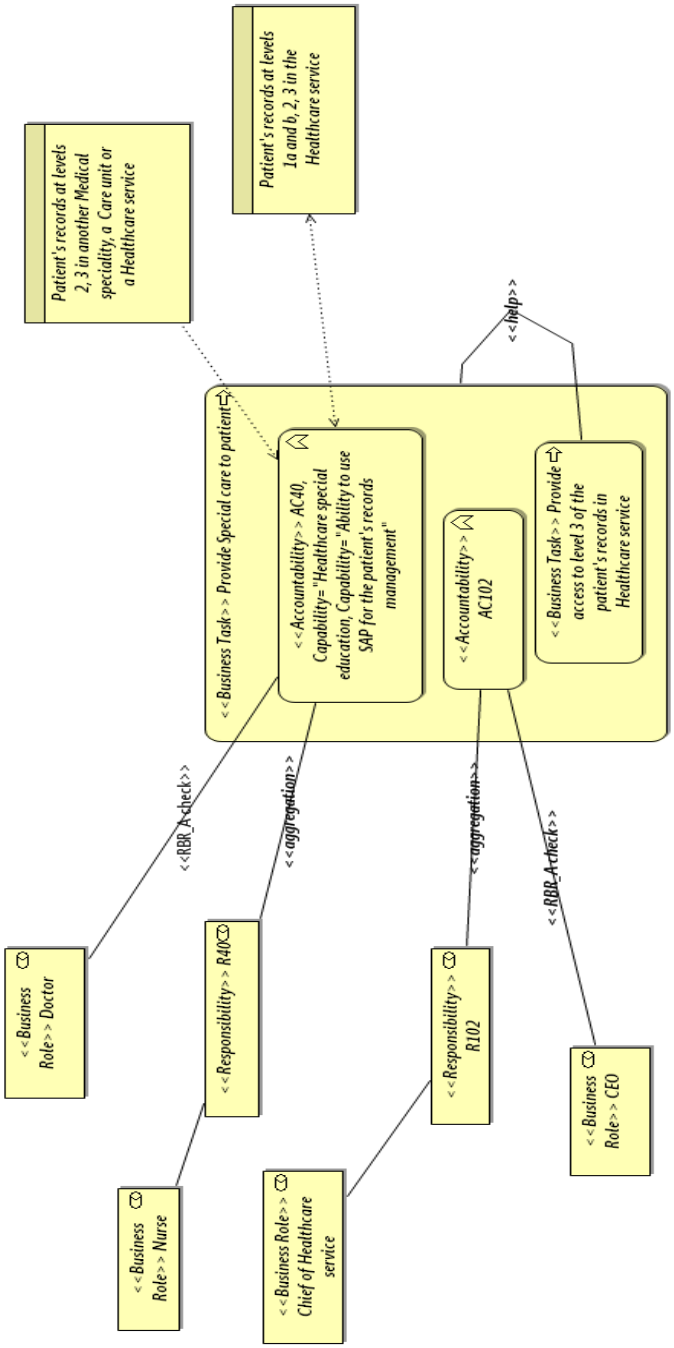


Figure C.5: Responsibilities *R40* and *R102*

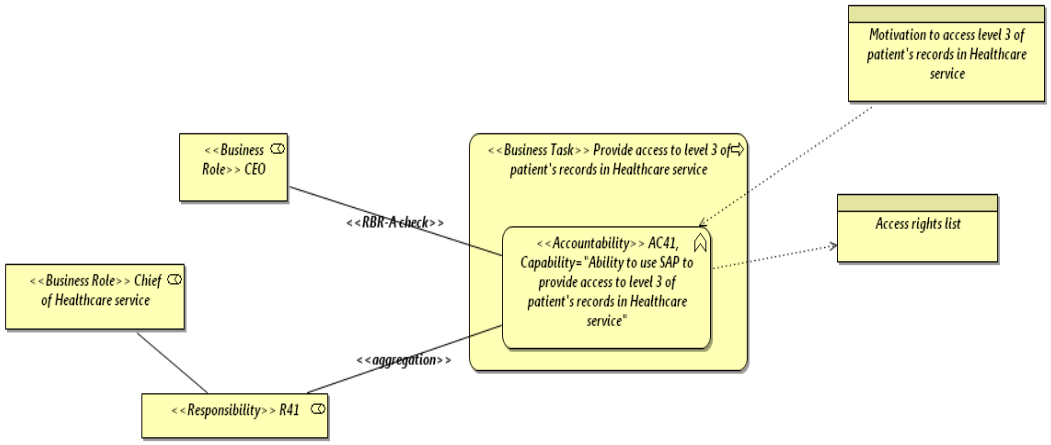


Figure C.6: Responsibility *R41*

C. LIST OF THE RESPONSIBILITIES FROM THE NURSES AND HEALTHCARE SPECIALISTS' SCENARII

Appendix D

List of the responsibilities from the quality analyst and statistician's scenarii

D. LIST OF THE RESPONSIBILITIES FROM THE QUALITY ANALYST AND STATISTICIAN'S SCENARII

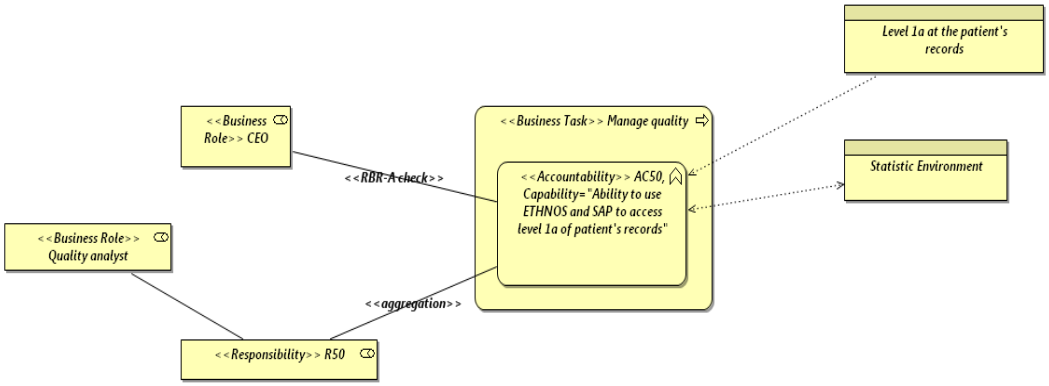


Figure D.1: Responsibility *R50*

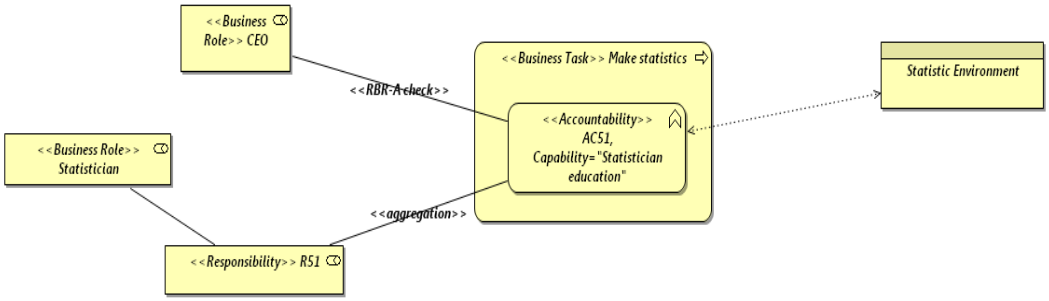


Figure D.2: Responsibility *R51*

Appendix E

RACI chart modelling with the Responsibility metamodel

COBIT is a framework for the governance and the security of the information system that provides a RACI chart which defines four types of specific responsibilities that cover most of the obligations existing in a company. This RACI chart also permits to associate business function to one or more of those four types of responsibilities. It defines the semantic of the responsibilities and permits to link them to the different tasks that are necessary to realise an activity (Section 3.3.1.1). Among those four responsibilities, responsible and accountable have already been addressed in the Responsibility metamodel.

E1 Responsible

Responsible for a task corresponds to the role which concretely realises the task. As a result, it is the actor who is assigned to the responsibility which aggregates the accountability to_do the procedure of the task. E.g., The `ChiefDoctor` is responsible to `HireEmployee`. This means that he is assigned to the `Responsibility` which aggregates the `Accountability` to_do the `HireEmployee Procedure` (Figure 4.12).

E2 Accountable

Accountable for a task corresponds to the role which gives the directions and authorisations to the performance of the task. As a result, this is the actor which is assigned to the responsibility which aggregates the accountability to_achieve the goal that defines the task. E.g., The `CEO` is assigned to the `Responsibility` which aggregates the `Accountable` to_achieve the `Goal` of `HireEmployee BusinessTask` (Figure 4.12).

E3 Consulted

Consulted for a task corresponds to the responsibility of the employees which provide information to the performance of the task. This task is modelled as a `StructuralTask` of `Advise` type. E.g., The `ITSpecialist` is responsible to `InstructHowToUseKnowledgeBase`. This means that he is assigned to the `Responsibility` which aggregates the `Accountability` to_do the `StructuralTask InstructHowToUseKnowledgeBase` that concerns the `BusinessTask SeekInfor- mationInPathologyKnowledgeBase`.

E4 Informed

Informed about the realisation of a task. According to our model of responsibility, being informed does not correspond to a particular responsibility. This is more a right which is afterwards necessary to perform another responsibility. E.g., the `Doctor` is responsible to `ReportAboutPathology`. This responsibility aims at producing a `Report` which is type of `BusinessObject` and the `ChiefDoctor` that `SuperviseTreatment` may (this is not represented on Figure 4.12) require a right to use this `BusinessObject`. In this case, the `ChiefDoctor` is informed according to COBIT RACI chart.

E5 RACI alternatives

Other responsibility matrix's have been proposed as a variant of the RACI chart:

Hightower (2008) proposes a RASCI model where the S completes the RACI chart with the responsibility to **Support**. This responsibility is played by actors that assist other actors who are responsible to_do the procedure of the task. This assistance is represented, in the Responsibility metamodel, by the statement that the actor that assists the performance of a task performs a sub–task of this task. E.g., The `Doctor` that `DoX-RayAnalysis` supports the `Doctor` that `AnalysePathology`.

Blokdijk and Menken (2008) propose a RACI–VS model where the V completes the RACI chart with the responsibility to **Verify** and S to **Sign**. The responsibility to verify is represented by the `Accountability` to_do the `StructuralTask` of the `Control` type and the responsibility to sign is represented by the `Accountability` to_do the `StructuralTask` of the `Approve` type.

Kendrick (2012) proposes the DACI model where D stands for **Driver** and A stands for **Approver**. The driver is responsible for steering the task although the approver is the one that makes most of the decisions. Both responsibilities may be represented as `Accountability` to_achieve a `Goal` in the Responsibility model. However, the driver responsibility aims to_achieve the `Goal` of a `Task` from a higher level and the approver responsibility aims to_achieve the `Goal` of `Task` which helps_for this `Task`.

Appendix F

Relationship between the concepts from the core and motivation ArchiMate metamodel

F. RELATIONSHIP BETWEEN THE CONCEPTS FROM THE CORE AND MOTIVATION ARCHIMATE METAMODEL

Figures F.1, F.2 and F.3 provide the relationships between the concepts from the core and motivation ArchiMate metamodel. The meaning of the types of associations is the following: (**a**)ccess, ass(**i**)gnment, (**c**)omposition, (**r**)ealisation, (**t**)riggering, a(**g**)gregation, ass(**o**)ciation, (**f**)low, (**s**)pecialisation, (**u**)sed by and i(**n**)fluence.

From ↓ / To →	Business Actor	Business Role	Business Collaboration	Location	Business Interface	Business Process	Business Function	Business Interaction	Business Event	Business Service	Business Object	Representation	Product	Contract	Meaning	Value
Business Actor	cfgostu	fiotu	fiotu	o	fiotu	fiotu	fiotu	fiotu	ot	iotu	ao	o	o	ao	o	o
Business Role	fiotu	cfgostu	cfgostu	o	cfgiotu	fiotu	fiotu	fiotu	ot	iotu	ao	o	o	ao	o	o
Business Collaboration	fgotu	cfgostu	cfgostu	o	cfgotu	iou	fiou	iou	ot	iotu	ao	o	o	ao	o	o
Location	io	io	io	cgos	io	io	io	io	io	io	io	io	o	io	o	o
Business Interface	fotu	fotu	fotu	o	cfgostu	ou	ou	ou	ot	iou	ao	o	o	ao	o	o
Business Process	fotu	fotu	fotu	o	fotu	cfgostu	cfgostu	cfgostu	ot	otu	ao	o	o	ao	o	o
Business Function	fotu	fotu	fotu	o	fotu	cfgostu	cfgostu	cfgostu	ot	otu	ao	o	o	ao	o	o
Business Interaction	fotu	fotu	fotu	o	fotu	cfgostu	cfgostu	cfgostu	ot	otu	ao	o	o	ao	o	o
Business Event	ot	ot	ot	ot	ot	ot	ot	ot	cgost	o	ao	o	o	ao	o	o
Business Service	ou	ou	ou	o	ou	ou	ou	ou	o	cfgostu	ao	o	o	ao	o	o
Business Object	o	o	o	o	o	o	o	o	o	o	cgos	o	o	cgos	o	o
Representation	o	o	o	o	o	o	o	o	o	o	or	cgos	o	or	o	o
Product	ou	ou	ou	o	ou	ou	ou	ou	o	gou	ao	o	cgos	agp	o	o
Contract	o	o	o	o	o	o	o	o	o	o	cgos	o	o	cgos	o	o
Meaning	o	o	o	o	o	o	o	o	o	o	o	o	o	o	cgos	o
Value	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	cgos
Application Component	fotu	fotu	fotu	o	fotu	fiotu	iou	iou	o	iotu	ao	o	o	ao	o	o
Application Collaboration	fotu	fotu	fotu	o	fotu	fiotu	fiotu	iou	o	iotu	ao	o	o	ao	o	o
Application Interface	ou	ou	ou	o	fotu	fotu	fotu	ou	o	ou	ao	o	o	ao	o	o
Application Function	ou	ou	ou	o	fotu	fotu	ou	ou	o	ou	ao	o	o	ao	o	o
Application Interaction	ou	ou	ou	o	fotu	fotu	ou	ou	o	ou	ao	o	o	ao	o	o
Application Service	ou	ou	ou	o	ou	ou	ou	ou	o	ou	ao	o	o	ao	o	o
Data Object	o	o	o	o	o	o	o	o	o	o	or	o	o	or	o	o
Node	ou	ou	ou	o	ou	ou	ou	ou	o	ou	aoru	o	o	aoru	o	o
Device	ou	ou	ou	o	ou	ou	ou	ou	o	ou	aoru	o	o	aoru	o	o
System Software	ou	ou	ou	o	ou	ou	ou	ou	o	ou	aoru	o	o	aoru	o	o
Infrastructure Interface	aou	aou	aou	o	aou	aou	aou	aou	o	aou	aou	o	o	aou	o	o
Network	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
Communication Path	o	o	o	o	o	o	o	o	O	o	o	o	o	o	o	o
Infrastructure Function	ou	ou	ou	o	ou	ou	ou	ou	O	ou	aou	o	o	aou	o	o
Infrastructure Service	ou	ou	ou	o	ou	ou	ou	ou	O	ou	aou	o	o	aou	o	o
Artifact	ou	ou	ou	o	ou	ou	ou	ou	O	ou	aor	o	o	aor	o	o
Junction	ft	ft	ft	o	ft	ft	ft	ft	ft	ft						

Figure F.1: Core concepts relations – part 1, **Source:** ArchiMate® 2.0 specifications (The Open Group (2012))

From ↓ / To →	Application Component	Application Collaboration	Application Interface	Application Function	Application Interaction	Application Service	Data Object	Node	Device	System Software	Infrastructure Interface	Network	Communication Path	Infrastructure Function	Infrastructure Service	Artifact	Junction
Business Actor	fot	fot	fot	fot	fot	o	o	o	o	o	o	o	o	o	o	o	ft
Business Role	fot	fot	fot	fot	fot	o	o	o	o	o	o	o	o	o	o	o	ft
Business Collaboration	fot	fot	o	fot	fot	o	o	o	o	o	o	o	o	o	o	o	ft
Location	io	io	io	io	io	io	io	io	io	io	io	io	io	io	io	io	io
Business Interface	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	ft
Business Process	fot	fot	o	o	o	o	o	o	o	o	o	o	o	o	o	o	ft
Business Function	fot	fot	o	o	o	o	o	o	o	o	o	o	o	o	o	o	ft
Business Interaction	fot	fot	o	o	o	o	o	o	o	o	o	o	o	o	o	o	ft
Business Event	ot	ot	o	o	o	o	o	o	o	o	o	o	o	o	o	o	ft
Business Service	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	ft
Business Object	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	
Representation	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	
Product	ou	ou	ou	ou	o	gou	ao	ou	ou	ou	ou	o	o	ou	gou	ao	
Contract	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	
Meaning	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o		
Value	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	
Application Component	cfgostu	cfgostu	cfgotu	ion	iou	iora	ao	o	o	o	o	o	o	o	o	o	ft
Application Collaboration	cfgostu	cfgostu	cfgotu	ion	iou	iora	ao	o	o	o	o	o	o	o	o	o	ft
Application Interface	fotu	fotu	cfgostu	ou	ou	iou	ao	o	o	o	o	o	o	o	o	o	ft
Application Function	ou	ou	ou	cfgostu	fot	oru	ao	o	o	o	o	o	o	o	o	o	ft
Application Interaction	ou	ou	ou	fotu	cfgot	oru	ao	o	o	o	o	o	o	o	o	o	ft
Application service	ou	ou	ou	ou	ou	cfgostu	ao	o	o	o	o	o	o	o	o	o	ft
Data Object	o	o	o	o	o	o	cgos	o	o	o	o	o	o	o	o	o	
Node	aou	aou	aou	aou	aou	aou	aou	cfgostu	cfgostu	cfgostu	cfgotu	o	o	iou	iou	aou	ft
Device	aou	aou	aou	aou	aou	aou	aou	cfgostu	cfgostu	cfgostu	cfgotu	o	o	iou	iou	aou	ft
System Software	aou	aou	aou	aou	aou	aou	aou	cfgostu	cfgostu	cfgostu	cfgotu	o	o	iou	iou	aou	ft
Infrastructure Interface	aou	aou	aou	aou	aou	aou	aou	fotu	fotu	fotu	cfgostu	o	o	ou	iou	aou	ft
Network	o	o	o	o	o	o	o	o	o	o	o	cgos	or		o	o	
Communication Path	o	o	o	o	o	o	o	o	o	o	o	o	cgos	o	o	o	
Infrastructure Function	aou	aou	aou	aou	aou	aup	aou	ou	ou	ou	ou	o	c	cfgostu	oru	aou	ft
Infrastructure Service	ou	ou	ou	ou	ou	ou	aou	ou	ou	ou	ou	o	o	ou	cfgostu	aou	ft
Artifact	oru	oru	oru	oru	oru	oru	aor	o	o	or	or	o	o	or	or	cgous	
Junction	ft	ft	ft	ft	ft	ft		ft	ft	ft	ft			ft	ft		ft

Figure F.2: Core concepts relations – part 2, **Source:** ArchiMate® 2.0 specifications (The Open Group (2012))

From ↓ / To →	Stakeholder	Driver	Assessment	Goal	Requirement	Principle	Constraint	Work Package	Deliverable	Plateau	Gap	Core Element	Business Actor	Business Role	Location	Value
Stakeholder	gaso	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Driver	0	gaon	on	on	on	on	on	0	0	0	0	0	0	0	0	on
Assessment	0	on	gaon	on	on	on	on	0	0	0	0	0	0	0	0	on
Goal	0	on	on	gaon	on	on	on	0	0	0	0	0	0	0	0	on
Requirement	0	on	on	ron	gaon	ron	gaon	0	0	0	0	0	0	0	0	on
Principle	0	on	on	ron	on	gaon	on	0	0	0	0	0	0	0	0	on
Constraint	0	on	on	ron	gaon	ron	gaon	0	0	0	0	0	0	0	0	0
Work Package	0	0	0	no	no	0	no	gaoft	no	no	0	no	noft	noft	0	0
Deliverable	0	0	0	no	no	no	no	0	gaso	0	0	go	go	go	0	0
Plateau	0	0	0	go	go	0	go	0	0	gaot	0	go	go	go	0	0
Gap	0	0	0	0	0	0	0	0	0	0	gaso	0	0	0	0	0
Core Element	0	0	0	no	no	no	no	0	0	0	0					
Business Actor	io	0	0	no	no	no	no	ioft	no	0	0					
Business Role	0	0	0	no	no	no	no	ioft	no	0	0					
Location	io	0	0	0	0	0	0	io	io	0	0					
Value	0	0	0	0	0	0	0	0	0	0	0					

Figure F.3: Motivation concepts relations, **Source:** ArchiMate® 2.0 specifications (The Open Group (2012))

Appendix G

European Court of Auditors case study

Introduction

In Chapter 5, we have mapped the Responsibility metamodel with the enterprise architecture framework ArchiMate. This integration permitted to improve the alignment between the business layer and the application layer of ArchiMate and allowed to enhance the provisioning of the access rights according to the employee's responsibilities. In the second part of the chapter, we have illustrated, based on a case study inspired by the Centre Hospitalier de Luxembourg, how using this ArchiMate integrated with ReMMo helps defining the responsibilities of the employees regarding the scenarii related to access the patient's records.

In Chapter 6, we have aligned the Responsibility metamodel with RBAC, and considering ArchiMate extended with the responsibility, we have optimised the assignment of permissions to the users, at the application layer, according to the responsibilities defined at the business layer. This enhancement of the application layer has also been illustrated at the Centre Hospitalier de Luxembourg for providing the employees assigned to the receptionist role with the access rights needed on specific professional software.

The objective of this appendix is to evaluate an intermediary version of the Responsibility metamodel. Based on this evaluation, ReMMo has been refined, some concepts have been added and others removed, and the associations between concepts sometime modified. This second case study illustrates how the Responsibility metamodel can help in defining the responsibilities of the employees from the European Court of Auditors and which highlights how the responsibilities can be used to improve the assignment of access rights to the employees as well. This case study has been performed at the court during fourteen months, from January 2011 to February 2012. During this period, twelve meetings were organised. Over the meetings, François Vernadat, Head of the Information Systems and Methods unit at the DIT, has explained the

context of the user provisioning and user account management, he has explained the enterprise architecture metamodel of the court and the IT infrastructure associate with the process as well.

As this case study is not based on the final version of the Responsibility metamodel like the one presented in Chapter 4, it is provided for information only.

At the European Court of Auditors, the management of identities is performed using the Oracle Identity Management (OIM) tool, which automatically provides the access rights for the employees to different applications such as Novell, Active Directory, or Lotus Notes. The court is currently running a project for increasing the automation of the Users' Accounts Management. In the meantime, this project concerns the upgrade of the OIM tool at the application layer, and the enhancement of the User Provisioning and User Account Management process, associated to the exploitation of this tool. During the case study, we have adapted the existing user's account management process, to consider the modification requested to fit the improved users' accounts management activities. Then we have modelled the responsibility of the employees involved in the process.

To understand, and to have a clear view over its information systems, the court has defined its own enterprise architecture metamodel. In the sequel of this appendix, we refer to this metamodel as the *ECA metamodel*. This metamodel includes business objects such as tasks, processes, roles and so forth. These business objects are necessary for the process modelling. An initial step in the case study was to analyse how ReMMo can be integrated with this ECA metamodel.

The appendix is structured as follows: Section G1 introduces the context of the case study at the European Court of Auditors. Section G2 presents the mapping of the Responsibility metamodel with the ECA metamodel, Section G3 introduces the User Provisioning and User Account Management process As–Is and To–Be, in Section G4 proposes the set of responsibilities to be assigned to the employees working on this process and finally, in Section G6, we evaluate the results of the case study.

G1 Context and objective of the case study

In this section, we introduce the European Court of Auditors, the Directorate General for Informatics and the objectives of the case study.

The European Court of Auditors

The European Union has a budget of approximately 120 billion euros, around 1 percent of the gross national income (GNI) of its 27 Member States. Compared to national budgets, this is a small share. However, for some Member States, funds from the EU play an important role in financing public activities and the total amount is close or equal to the GNI of some Member States. The revenue of the European Union mainly consists of contributions from Member States based on their gross national income (GNI – 65.4 percent) and on a measurement connected to value added tax collected by the Member States (VAT – 16.9 percent). Customs and agricultural duties (so called traditional own resources – 16.5 percent) also represent a significant share of

revenue. The composition of the budget has evolved over time, agriculture and cohesion policies being its major components.

The budget is decided annually by the Council and the Parliament within the context of seven–year financial frameworks, i.e., representatives of the Members States, and the directly elected European Parliament. The European Commission proposes the budget and is also responsible for implementing it. A very significant proportion – notably agricultural and cohesion spending – is implemented in cooperation with the Member States. Depending on the spending schemes, national administrations may be responsible for setting spending strategies, selecting beneficiaries and projects and making payments. A specific feature of Commission expenditure is the high percentage of payments based on claims submitted by the beneficiaries themselves, be they farmers or project managers throughout the Union.[1]

The European Court of Auditors is the EU Institution established by the treaty to carry out the audit of EU finances. As the EU's external auditor it contributes to improving EU financial management and acts as the independent guardian of the financial interests of the citizens of the Union. The court renders audit services through which it assesses the collection and spending of EU funds. It examines whether financial operations have been properly recorded and disclosed, legally and regularly executed and managed so as to ensure economy, efficiency and effectiveness. The Court communicates the results of its audits in clear, relevant and objective reports. It also provides its opinion on financial management issues[2]

The DIT[3] is the department of the court responsible for the management of information technology. The DIT is composed of three units: User Services and Operations, Information Systems and Methods, and IT Office Library.

Directorate–General for Informatics

The Directorate General for Informatics (or "DIGIT") is a Commission service based in both Brussels and Luxembourg which has for mission the definition of IT strategy for the Commission and the provision of a modern, high–performance information technology and telecommunication infrastructure.

The role of the DIGIT is to manage and coordinate the Commission's IT and telecommunications resources on behalf of the Commission's services and, in particular to formulate and implement a dynamic global IT strategy supporting the Commission's priorities[4].

The main services provided by DIGIT are:

- Corporate information systems to support the business processes of the Commission, in partnership with the other directorates–general and services. These systems are principally, but not exclusively, in the field of document management, financial management, planning and reporting, and human resources systems.

[1] http://eca.europa.eu/portal/page/portal/aboutus/TheCourtsroleandwork
[2] This paragraph is extracted from http://eca.europa.eu/portal/page/portal/aboutus/abouttheeca
[3] DIT is the French acronym for Direction des Technologies de l'Information
[4] http://ec.europa.eu/dgs/informatics/about/

- A Commission–wide, secure, reliable and high–performance information technology and telecommunication infrastructure to support the Commission's activities and to enable the implementation of the e–Commission.
- Consulting services to promote best practice in the application of modern information and telecommunications technology, including OSS technologies.

The European Court of Auditors and the DIGIT closely cooperate on some information systems. For instance, the court and the Commission use the same human resource management system (SYSPER2) or the same professional training system (SYSLOG formation)

Objectives of the case study

At the European Court of Auditors, the assignment of employees to tasks is realised using the intermediary of roles. This approach is common in large companies, well proven, and fully justified as well.

The first problem, in the court, is when the tasks of a process need to be assigned to the employees, the process does not define their final accountability. For instance, who is accountable to perform, or who is accountable to decide upon, regarding the realisation of the task. It only describes which role realises the tasks. The second problem is that of the process description. It is also neither defined what the required capabilities of the employees are nor what access rights they need.

Recognising this lack of information and the impact that it represents on the management of access rights, the court has decided to investigate how using the intermediary concept of responsibility could contribute to solving these problems.

Therefore, the case study has two objectives. For this research, the case study evaluates how the Responsibility metamodel can help to define the responsibilities of the employees at the European Court of Auditors, and it highlights how the responsibilities modelled can be used to improve the assignment of access rights to these employees. The case study also has for objective to propose a solution for the court to refine the provisioning of the access rights to its employees.

In the following sections, we successively present the integration of the Responsibility metamodel with the ECA metamodel, introducing the User Provisioning and User Account Management process As–Is and To–Be, and finally, we propose a set of responsibilities to be assigned to the employees working on this process.

G2 Integration of the ECA metamodel with the Responsibility metamodel

In this section, we integrate the ECA metamodel with ReMMo. Therefore, we use the three steps approach defined by Michaël Petit in Petit (2003): the first step is the preparation of the integration, the second step is the investigation and the definition of the correspondences and the third step is the integration of both metamodels.

The court enterprise architecture metamodel

To support the management of its information system, the European Court of Auditors makes use of a dedicated architecture framework named CEAF[1]. The particularity of the CEAF is that it is business oriented and only provides a framework for the business entities in relation with IT usage. Considering the business in the heart of the framework allows continual business/IT alignment. In complement to its four layers, the CEAF also contains a set of architecture standards that gather methods, vocabulary and rules to comply with. Such a rule is, for instance, at the business layer, *the DIT needs to understand the business activities to automate them.*

The DIT has defined its own enterprise architecture metamodel (Figure G.2) based on the CEAF. This metamodel is formalised using an entity–relationship model and is computerised using the Corporate Modeler Suite from CaseWise[2]. It is elaborated around the four vertical layers (Figure G.1) as the ones that compose the CEAF and each of them represents, accordingly, a perspective in the architecture:

- **The business layer**. This layer aims at formalising the main business processes of the organisation (i.e., process map and process flows in terms of activities).
- **The functional layer**. This layer defines the views needed to describe the business processes in relation to functions and services.
- **The application layer**. This layer describes the IT applications or information systems and the data exchanges between them.
- **The data layer**. This layer describes the IT infrastructure in terms of its servers, computer networks, devices, security devices, and so forth.

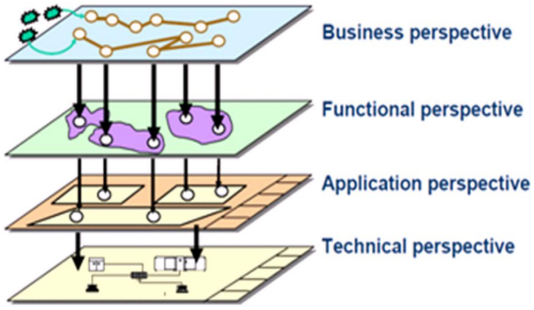


Figure G.1: Four layers of the CEAF

Each one of the four layers includes a set of concepts, significant for the layer, and may contain different types of views. Each view is based on a template diagram. In Sections G3 and G3, the identity management process is described using the specific court *Template diagram 14 – BA– Activities BPMN.*

The court enterprise architecture metamodel is documented in French in Casewise. Figure G.2 has been kept in French, but it has been translated in UML and into English in Figure G.3.

[1]CEAF is the French acronym for Cadre d'Architecture d'Entreprise de la Commission (européenne), which means, in English: European Commission Enterprise Architecture framework

[2]http://www.casewise.com/products/modeler

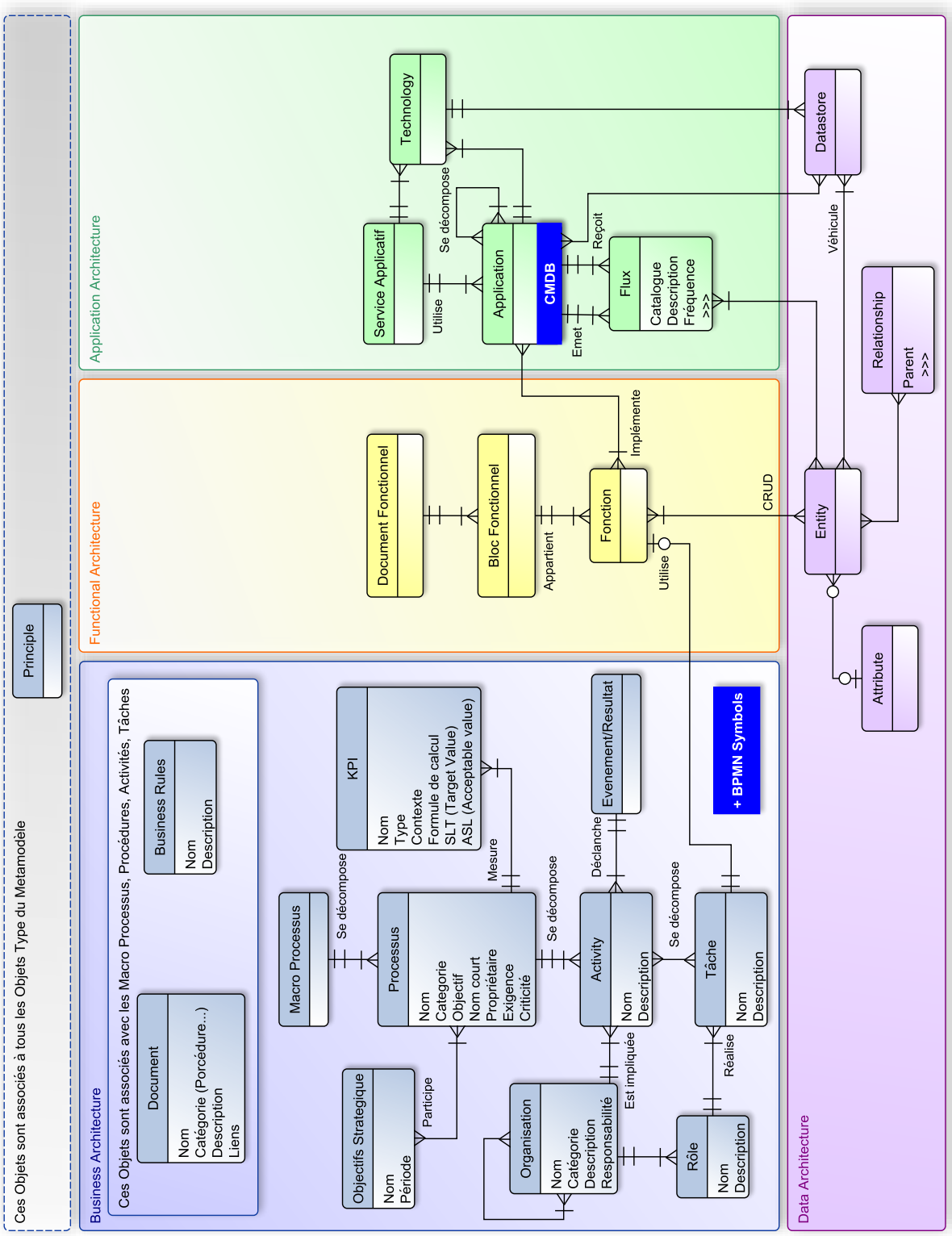


Figure G.2: Court main Enterprise Architecture metamodel

Preparation for the integration

As defined in Petit (2003), this first step of the integration of both metamodels requires a preparation of the integration. Therefore, an integration strategy is defined and provides the baselines for the integration such as the context of the integration which has been presented in the introduction of this chapter, the selection of a common language for the representation of the metamodels to be integrated, the selection of the appropriate subset of concepts represented in the integrated metamodel, and so forth.

Subset of concepts concerned by the integration

This activity of selecting the appropriate subset of concepts considered for the integration has been added to the method of Petit (2003) and is required to address the concepts from the metamodels that are meaningful for the assignment of tasks to the employees and for the definition of the rights and capabilities required therefore.
The subset of concepts concerned by the integration, in the ECA metamodel, includes:

- **The concept of role**. This concept is used to represent the notion of executing a task of a process. It is associated to the concept of a task that it realises and to the concept of organisation to which it belongs.
- **The concept of task**. This concept is used to describe how the activities are performed. Task is carried out by a single actor (not represented in the ECA metamodel), is performed continuously and can not be interrupted. The task is associated to the concept of role which it realises, to the concept of activity that it composes, and to the concept of function that it uses.
- **The concept of function**. This concept enables to break–down an information system in functional domains, functional blocks and functionality items. A function block is defined by the business concepts it manages on behalf of the SI, combining the functions (functions related to business objects), and production rules of the data that it communicates. It is associated to the concept of task, of information system (the application) that implements it and of the entity that it accesses in a CRUD mode (Create, Read, Update, Delete).
- **The concept of entity**. This concept represents the business data conveyed by the information system or handled by an application. In the latter case, we speak of information data. It means that the physical data model implemented is not described in systems/database. The entity is accessed by the function, is associated to flow, is defined by attributes and relationships and is stored in datastore.
- **The concept of application**. This concept represents a component that contributes providing a service to a dedicated business line or for a particular system. Regarding the association of the concept of application with other concepts: the application is used by the application service, is composed of one or more other application(s), uses a technology, sends and receives flux and implements functions.

In the Responsibility metamodel, we keep the following concepts:

- The concept of responsibility
- The concept of business role

- The concept of business task
- The concept of right
- The concept of capability
- The concept of accountability
- The concept of employee

Selection of a common representation language

For the integration, UML was used because it is accurate enough for our purpose, standard and commonly used. As a consequence, the ECA metamodel is formalised using the entity–relation model and has been translated in UML class diagram on Figure G.3

Investigation and definition of the correspondences

In Petit (2003), the author explains that this second step analyses the correspondences between the classes of the metamodels. Those correspondences exist if correspondences among instances of these classes taken two by two can be generalised. Therefore, it is advisable to carry out one or more case study(ies) to model real world elements with both languages and, to compare the semantics of the obtained models.

This second step analyses the correspondences between classes of the metamodels. Those correspondences exist if correspondences among pair of classes exist and if correspondences between instances of these classes, taken two by two, can be generalised. The latter are analysed through the case study. The ECA metamodel and the Responsibility metamodel have three correspondences between their classes.

The correspondence between the UML associations in the ECA metamodel and in ReMMo is also analysed during the integration of both metamodels. As only two classes from the ECA metamodel correspond to three classes from the Responsibility metamodel, it results that there does not exist a correspondence between associations of classes from both metamodels.

Correspondences between the classes: The ECA metamodel and the Responsibility metamodel have three exact correspondences between their classes:

- **Role from the ECA metamodel and business role from the Responsibility metamodel**. The concept of role from the ECA metamodel is represented in the business architecture. It is an element that belongs to the organisation and that realises business tasks. Hence it reflects a business role rather than an application role and as a result it corresponds to the business role of the Responsibility metamodel.
- **Entity from the ECA metamodel and information from the Responsibility metamodel**. The concept of entity from the ECA metamodel is equivalent to the concept of information from ReMMo. Both concepts are accessed by a human or an application component and specific access rights are necessary to access them.
- **Task from the ECA metamodel and business task from the Responsibility metamodel**. The concept of task from the ECA metamodel and the concept of business

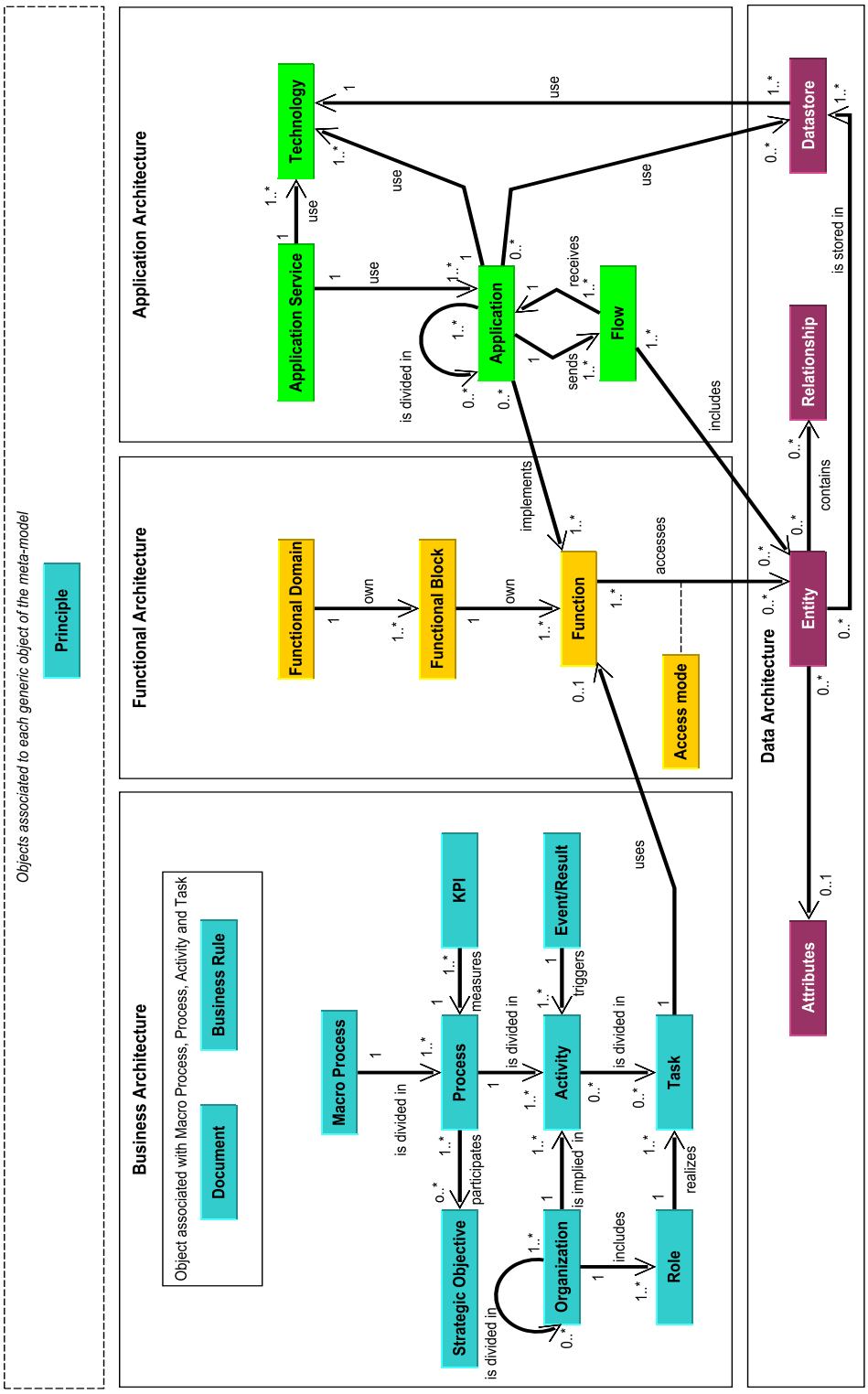


Figure G.3: Court main Enterprise Architecture metamodel in UML diagram

task from the Responsibility metamodel semantically have the same meaning. The task from the ECA metamodel includes the business architecture and it corresponds to a task performed on the business side. Therefore, it is equivalent to the task from ReMMo. According to the definition from the ECA, we note that the task is performed by a single actor. This is a constraint which does not exist in the Responsibility metamodel and which needs to be considered in the integration step.

Integration of the Responsibility metamodel with the ECA metamodel

The third step defined in Petit (2003) corresponds to the integration of the metamodels. During the analysis of the correspondences between the concepts and the UML associations between these concepts, we have observed some minor differences. Despite the influence of these differences, in order to consider that a sufficient correspondence exists between the elements and in order to consider them during this third step of integration, we have to analyse this difference in depth and formalise the integration rules to consider having a perfect integration.

To construct an integrated metamodel that enriches the ECA metamodel with the Responsibility metamodel, we have, like in Chapter 5, to define a set of integration rules. Like for the integration of the Responsibility metamodel and the ECA metamodel, we decide that (1) when a correspondence exists between one class from the ECA metamodel and one class from ReMMo, we preserve the name of the class from the ECA metamodel, (2) when the class of the Responsibility metamodel has no corresponding class in the ECA metamodel, this class is integrated in the integrated metamodel and it preserves its name from ReMMo, (3) when a correspondence exists with conflicts between the definition of the classes, the classes are integrated in the integrated metamodel, we preserve the name of the class from the ECA metamodel and we additionally include integration rules to be respected in case of using the integrated metamodel. Finally, (4) when classes are different in both metamodels, in each case we motivate our integration preferences.

Classes integration:

1. **Classes that correspond exactly**
   - The role from the ECA metamodel and the business role from the Responsibility metamodel
   - The entity from the ECA metamodel and the information from the Responsibility metamodel
2. **Classes that only exist in the Responsibility metamodel**
   - Employee
   - Responsibility
   - Capability
   - Accountability
   - Rights and the type of right: the right to use
3. **Classes that only exist in the ECA metamodel**

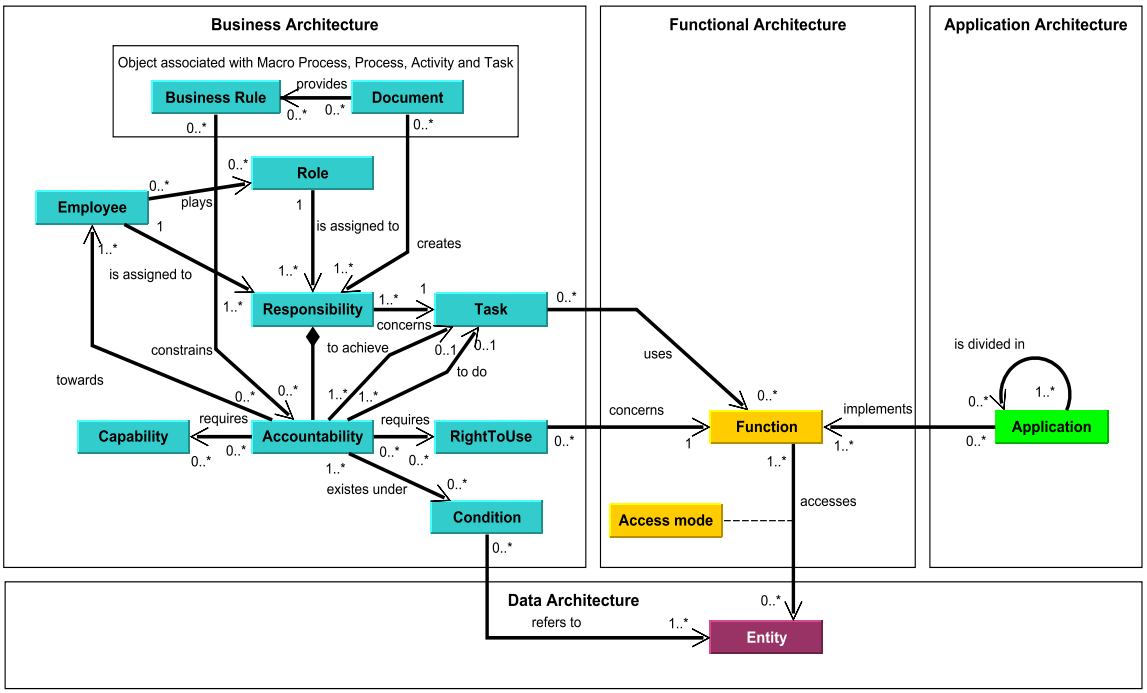


Figure G.4: Integrated ECA metamodel and the Responsibility metamodel

- Function

4. **Classes that correspond under integration rules**

   - **The business task from the Responsibility metamodel and the task from the ECA metamodel:**
     In the ECA metamodel, a task is performed by a single actor. The ECA metamodel description does not define the granularity level of a task and, for instance, it does not define if "doing a task", "advising for the performance of a task", "making decisions during the realisation of a task" are considered as three tasks or as a single task. In the first case, three actors may be assigned separately to each of the three propositions although, in the latter, only one actor is assigned to it. In ReMMo, many employees may be assigned to many responsibilities regarding a business task. We observe that, in practice, this is most often the case and, as explained in Section G3, this is also identical to the court.

     Therefore, in our integrated metamodel, we consider that a task may be concerned by more than one accountability. This latter composes the responsibility assigned to one or more employee(s).

     For instance, if we consider the task to deploy a new software component on the court network, we have a first responsibility to effectively deploy the solution. This responsibility is assigned to an IT administrator who is accountable to the manager of

his unit. This means that he must justify the realisation or not of the deployment and that he is positively or negatively sanctioned by the unit manager. The unit manager, for this deployment, is responsible for making the right decisions, for instance, to decide the best period of the day for the deployment, to give the go/no go after the deployment test, and so forth. This responsibility is directly handled by the unit manager that must justify his decision and is sanctioned accordingly by his own superior, for instance, the department manager, and so forth. This illustration explains how many responsibilities may be concerned by a single task.

5. **Classes that are different in the Responsibility metamodel and in the ECA metamodel**

   - **The access rights class from the Responsibility metamodel and the access mode class from the ECA metamodel:**

     The access rights class corresponds to a type of right in the Responsibility metamodel and corresponds to an access mode in the ECA metamodel. In the ECA metamodel, the entity is accessed by the class of function which, additionally, is associated to a task and application of the IS that implements it. As a result, the access rights is already considered in the ECA metamodel, but it is directly associated to the task class by the intermediary of function.

     In the integrated metamodel, we retain the class of function which is interesting to consider, in the meantime, classes from the business architecture, classes from the application architecture and classes from the data architecture. However, to restrict the usage of a function only for what is strictly necessary, we do not consider that it is associated to a task, but that its usage is a type of rights required by a responsibility and necessary for an accountability.

     As such, an employee with the accountability of doing a task gets the rights to use a certain function, an employee with the accountability of deciding about the execution of a task gets the rights to use another function, and so forth.

UML associations integration:

1. **UML associations from the Responsibility metamodel which complete or replace, in the integrated metamodel, the UML associations from the ECA metamodel:**

   - The direct UML association between a role and a task is replaced by “a role is composed of responsibilities, themselves composed of accountabilities concerning task”.
   - The UML association between the task and the function it uses is replaced by the UML association “an accountability concerning a task necessitates right(s)” and “one type of right is the right to use a function”

2. **New UML associations from the Responsibility metamodel, that do not exist in the ECA metamodel, and which are integrated in the integrated metamodel:**

   - The responsibility requires capability

- The responsibility requires right
- The employee is assigned to one or more responsibility(ies) and to one or more role(s)
- The capability is necessary for a task
- The right is necessary for a task

G3 User Provisioning and User Account Management process evolution

The user provisioning concerns the providing, adapting or removing of access rights when a newcomer arrives, or when an employee changes status or department or leaves the court.

The management of the users' identity and access rights are areas in which the DIT invests considerably. Indeed, since each employee of the court needs different access rights on the information system, these access rights must be accurately provided according to their profile. Therefore, to manage these rights, the DIT has invested in the Oracle Identity Management (OIM) tool. This tool is central to the users' accounts management activity and, as illustrated in Figure G.5, is connected, on the one hand, to the applications that provision the user profile (COMREF and eAdmin) and, on the second hand, to the user directories that provision access rights rules (Active Directory, Lotus Notes, and so forth).

COMREF is the central human resource database of the European Commission and is a subset of Sysper2[1]. The COMREF database is located in Brussels and gathers a set of official and employees information such as the type of contract, occupation, grade, marital status, date of birth, place of work, department, and so forth. This information is synchronised, each day, with COMREF_ECA[2] and with the OIM tool. In parallel, additional information is also uploaded in the OIM tool for the subset of data relative to court workers (employees or external staff), directly from the court, e.g., the office number, the entry ID card, the phone numbers, the telephone PIN code, and so forth. This information is also synchronised, each day, with the central COMREF database.

At the business layer, processes have been defined to support the activities of the employees who manage (such as the administrators) or who use the system (such as the secretaries who fill in the fields related to the PIN code or phone number). The case study focusses on one of the processes which is the users' account management process. The court's users' accounts management process aims at defining an ordinate set of tasks to manage the request, the creation, issue, suspension, modification or closure of users' accounts and to accordingly provide the employees with a set of users' privileges. More specifically, in the case study, we depict the evolution of this process after an improvement of the automation of the OIM tool provisioning with COMREF database, and we define the responsibilities of the employees' involved in this process.

[1]Sysper2 is the Human Resource Management solution of the European Commission that supports the personnel recruitment, career management, organisation chart, time management

[2]COMREF_ECA is a dedicated mirror of COMREF for the officials and employees of the court in Luxembourg

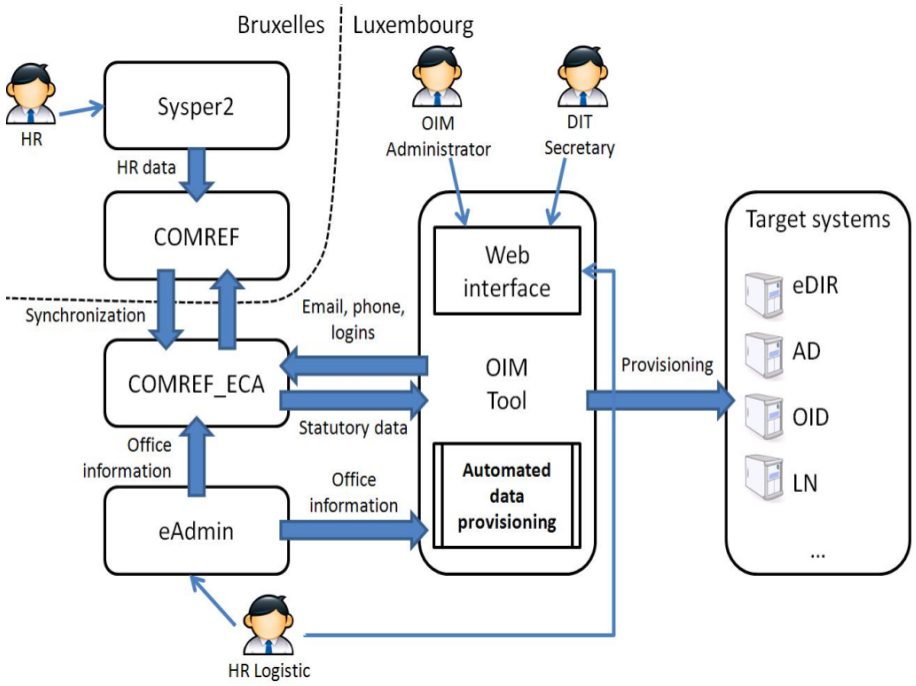


Figure G.5: ECA OIM overview

Therefore, in Section G3, we review the process before the improvement (referred to as As–Is) and after the improvement (referred to as To–Be). The description of the As–Is process in Section G3 explains how the process existed before the automation. The process is explained in Figure G.6. The description of the To–Be process is described in Figure G.7.

As–Is User Provisioning and User Account Management process

The As–Is process begins with a daily Sysper2 extraction. This extraction is used to identify the newcomers and the status change of employees, such as name change, end data contract change, or modification of department assignment. According to the change, a precise set of tasks is defined. For instance, when a newcomer arrives at the court, the DIT secretary identifies phone and fax numbers, provides a PIN code, and sends the information to the OIM administrator. The OIM administrator adds this information to the OIM tool, as well as a PERID (unique personal ID provided by the HR) and a job category (retrieved in Electronic or Paper "Information note" also from the HR). This information is, afterwards, automatically provisioned in the target applications. In parallel, the HR unit attributes an office using the eAdmin tool and some application's owners modify and add additional user information in the software that they administer (this is mainly the case of Active Directory and Novell).

For the other changes, the As–Is process describes how the OIM administrator is solicited along the process and according to the changes:

- for validating the name with the user management policy document and modifying the name in the OIM tool in case of name change,

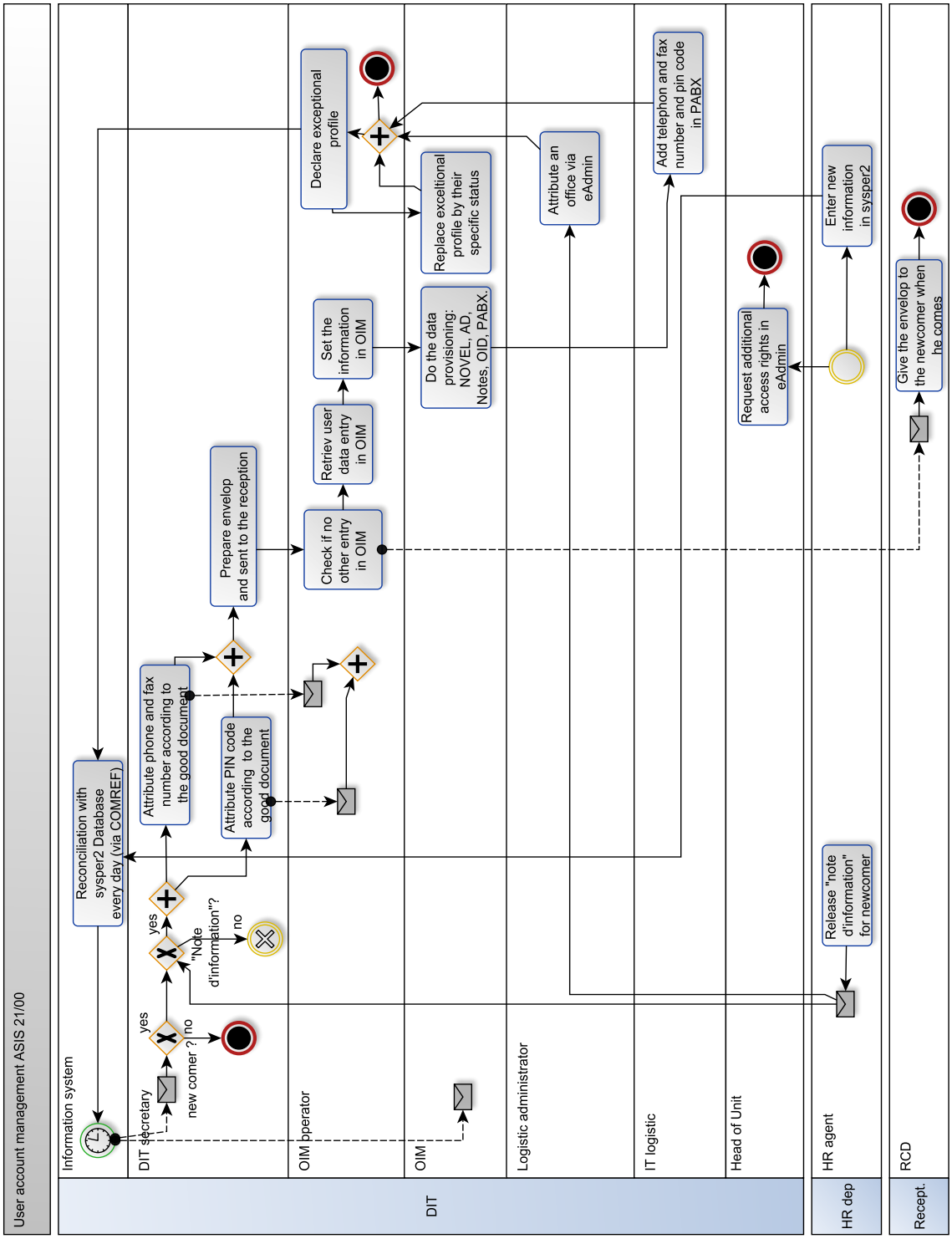


Figure G.6: *User Provisioning and User Account Management* process As–Is

- for modifying the service in the tool in case of service change,
- for changing the users' contract end date in case of end date modification in the contract,
- and so forth.

To–Be User Provisioning and User Account Management process

Since the court is a European administration with more than one thousand officials and employees, the enhancement of the User Provisioning and User Account Management process is realised stepwise, and progressively. In the frame of the case study, the targeted enhancement concerns the automatic provisioning of the OIM tool with information retrieved from COMREF. This enhancement is aimed at reducing the information handling and re–encoding from COMREF to the OIM tool, thereby, avoiding risks of errors.

In parallel to the application modification resulting from this enhancement, the users' account management process has also been improved, and the responsibility of the stakeholders modelled. Due to the improvement, the adaptations of the process, like for instance for a newcomer, were the following: the DIT secretary still needs to identify phone and fax numbers, and PIN code, and send the information to the OIM administrator. The OIM administrator still introduces the information coming from the DIT secretary in the OIM tool, as well as the PERID, but he does not have to manually encode other information such as name, end contract data, service and so forth. The HR Logistic unit still has to attribute an office using the eAdmin tool.

The tasks of the OIM administrator are also facilitated when changes occur regarding the users' attributes. In this case, all the manipulations are handled by the OIM tool.

G4 Responsibility modelling and assignment

Approach to model and assignment of the responsibilities

In this section, we propose a sequence of four steps to model the responsibilities of the employees involved in the upgraded users' accounts management process.

1. **Identification of business tasks**. The business tasks are defined in ReMMo in Figure 4.2 and exist in the integrated ECA–Responsibility metamodel as the concepts of tasks (Section G2). In Figure G.7, the process is modelled in BPMN and the tasks are represented by rectangles.

   In this step, we identify the tasks for which we have to define responsibilities, though we do not consider the tasks that are performed by an application component and for which defining the responsibility is inappropriate according to our definition of the responsibility in Section 4.5.1. After the process enhancement, six tasks are remaining. Those task are, i.e., Release "Note d'information[1]", Complete Sysper2, Attribute an office using eAdmin,

[1]Information note in English

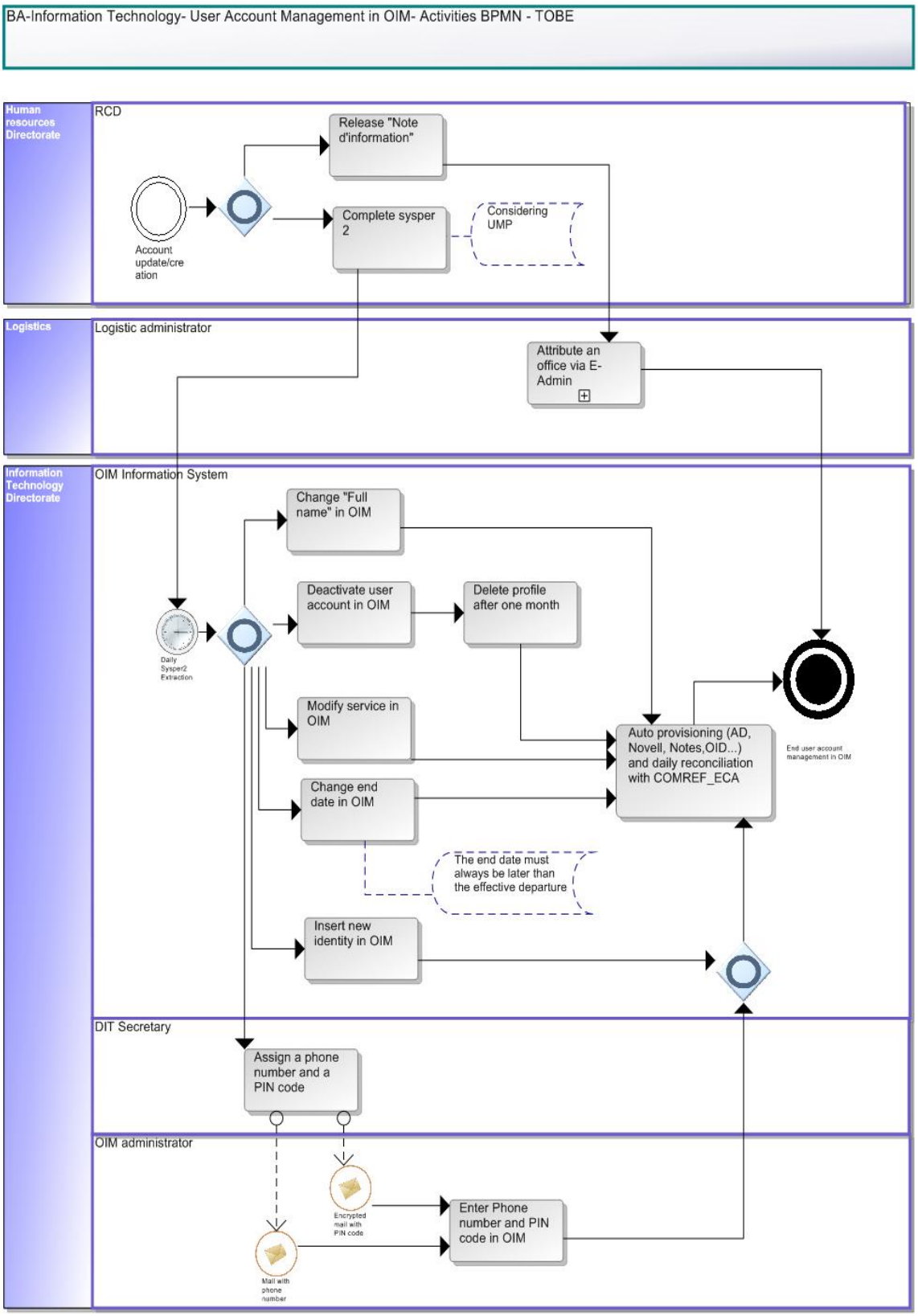


Figure G.7: *User Provisioning and User Account Management* process To–Be

Assign a phone number and a PIN code, Enter phone number and PIN code in OIM, Auto provisioning and daily reconciliation.

2. **Identification of the responsibilities and accountabilities**. The accountability, as explained in Section 4.5.1, defines which obligation(s) compose(s) a responsibility for a business task and which justification is expected. In the ECA metamodel, this concept of accountability has been preserved (Figure G2) since it is important to distinguish what really are the accountabilities of the court's employees regarding the business tasks.

   In this step, we have reviewed, for each of the tasks, the existing accountabilities for each of the responsibilities. Primarily three of them have been retained. The obligation to "Do" which composes the responsibility of performing the task, the obligation to "Decide about" which composes the responsibility of being accountable for the performance of a task and the obligation to "Advise" which composes the responsibility to give advice for the performance of the task. I.e., three types of accountabilities concern the task "Assign a phone number and a PIN code" and the task "Attribute an office using eAdmin".

3. **Identification of the rights and capabilities**. The rights and capabilities are elements required by a responsibility and necessary to achieve accountabilities (Figure 4.8). Both concepts have been introduced in the integrated metamodel in Figure G2.

   In this step, we have analysed, accountability by accountability, which capabilities and rights are necessary to realise what accountability. In the integrated ECA–Responsibility metamodel, the access rights (which are type of a right) are no longer directly associated to the realisation of an action regarding an information (like, read a file), but are rights to use a function that realises an action (e.g., CRUD) regarding an entity and the use of an application that manipulates this entity. E.g., the Responsibility *OIM 7* (Table G.7) is assigned to Josy Schilder who requires to use the function that realises Read in eAdmin.

4. **Assignment of the responsibilities to the employees**. As far as the responsibilities are modelled, we may assign them to the employees, considering their roles in the organisation. As explained in Figure 4.5, a responsibility may be assigned directly to an employee or to a role.

   In the court case study, some responsibilities are directly assigned to employees and others are assigned to role. For instance, the Responsibility *OIM 1* (Table G.1) is composed of the accountability to do the task Release "Note d'information". This responsibility is assigned to the role Human Resources Directorate/RCD (recruitment career development), although the Responsibility *OIM 10* (Table G.10) is composed of the accountability to do the task "Enter Phone number and PIN code in OIM" and is assigned directly to the employee Francis Carambino.

Within the User Provisioning and User Account Management process, six tasks have been identified and have generated thirteen responsibilities. The task Release "Note d'information" is concerned by two accountabilities: the accountability to do the task and the accountability to decide about the achievement of it. All the employees of the role Human Resources Directorate/RCD (recruitment career development) may Release "Note d'information" but only Gerald Hadwen makes decisions regarding it (Tables G.1 and G.2).

Responsibility *OIM 1*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Do	Release "Note d'information"	All	Gerald Hadwen	
Role	**Backup role**	**Right**	**Capability**	
Human Resources Directorate/RCD (recruitment carrier development)	RCD Unit Chief			

Table G.1: Responsibility *OIM 1*

Responsibility *OIM 2*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Decide about	Release "Note d'information"	Gerald Hadwen		
Role	**Backup role**	**Right**	**Capability**	
RCD Unit Chief				

Table G.2: Responsibility *OIM 2*

The task "Complete Sysper2" is concerned by two accountabilities: the accountability to do the task and the accountability to verify its achievement. Samantha Doherty is assigned to the realisation of the task and the achievement is verified by one of the employees with the role RH Manager. We note that Samantha Doherty has the rights to Read–Write–Modify data in Sysper2 although the verifying employee only needs the right to Read the input in Sysper2 (Tables G.3 and G.4).

Responsibility *OIM 3*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Do	Complete Sysper2	Samantha Doherty	Garry Gehring	
Role	**Backup role**	**Right**	**Capability**	

Human Resources Secretary	Human Resources Directorate	Read–Write–Modify access to all Sysper2 functions and Read access to RETO (REservation TOol)[1]	Sysper2 and SQL training	

Table G.3: Responsibility *OIM 3*

Responsibility *OIM 4*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Verify	Complete Sysper2		Garry Gehring	
Role	**Backup role**	**Right**	**Capability**	
RH Manager	Human Resources Directorate	Read access to all Sysper2 functions	Sysper2 training	

Table G.4: Responsibility *OIM 4*

The task "Attribute an office using eAdmin" is concerned by three accountabilities: the accountability to do the task, the accountability to decide about its achievement, and the accountability to advice about its realisation. Barbara Smitz is assigned to the realisation of the task, has Antonio Sanchis for backup and is answerable towards the latter and towards Reynald Zimmermann. Steven Jaatun may be consulted by Barbara Smitz and has the accountability to advise her. Both, Antonio Sanchis and Steven Jaatun need, at least, a Read access to eAdmin, although, Barbara Smitz needs a Read–Write access (Tables G.5, G.6 and G.7).

Responsibility *OIM 5*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Do	Attribute an office using eAdmin	Barbara Smitz	Antonio Sanchis/ Reynald Zimmermann	Antonio Sanchis

[1] RETO is a personal numbers (like PERID) booking tool

Role	**Backup role**	**Right**	**Capability**	
Logistic administrator	Logistic Head of Unit	Read–Write access in eAdmin	eAdmin manipulation training	

Table G.5: Responsibility *OIM 5*

Responsibility *OIM 6*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Decide about	Attribute an office using eAdmin	Antonio Sanchis		
Role	**Backup role**	**Right**	**Capability**	
Logistic Head of Unit		Read access in eAdmin	eAdmin manipulation training	

Table G.6: Responsibility *OIM 6*

Responsibility *OIM 7*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Advise about	Attribute an office using eAdmin	Steven Jaatun		
Role	**Backup role**	**Right**	**Capability**	
Logistic		Read access in eAdmin	eAdmin manipulation training	

Table G.7: Responsibility *OIM 7*

The task "Assign a phone number and a PIN code" is concerned by two accountabilities: the accountability to do the task and the accountability to decide and to advice about its achievement. Maria Dos Sanchez is assigned to the realisation of the task and she has two backups: Jeff Vaillant and Francis Carambino. In addition, Francis Carambino has to decide and advise her/them about the realisation of the task and he has, therefore, Philippe Melvine for backup (Tables G.8 and G.9).

Responsibility *OIM 8*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Do	Assign a phone number and a PIN code	Maria Dos Sanchez	Nizar Simon	Jeff Vaillant/ Francis Carambino
Role	**Backup role**	**Right**	**Capability**	
DIT secretary	Director Information Technology	Read–Write file XXX in Excel	Excel manipulation training	

Table G.8: Responsibility *OIM 8*

Responsibility *OIM 9*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Decide about/advise	Assign a phone number and a PIN code	Francis Carambino	Nizar Simon	Philippe Melvine
Role	**Backup role**	**Right**	**Capability**	
OIM Administrator	Director Information Technology	Read file XXX in Excel	Excel manipulation training	

Table G.9: Responsibility *OIM 9*

The task "Enter phone number and PIN code" is concerned by two accountabilities: the accountability to do the task and the accountability to decide about its achievement. Francis Carambino is assigned to the realisation of the task and he has for backup: Philippe Melvine. Marco Jonhson has to decide about the realisation of the task and he has, therefore, the right to Read OIM Report (Tables G.10 and G.11).

Responsibility *OIM 10*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Do	Enter phone number and PIN code in OIM	Francis Carambino	Marco Jonhson	Philippe Melvine

Role	**Backup role**	**Right**	**Capability**	
OIM Administrator	IAM[1] Service Manager	Read–Write access to OIM tool	OIM administration training	

Table G.10: Responsibility *OIM 10*

Responsibility *OIM 11*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Decide about	Enter phone number and PIN code in OIM	Marco Jonhson		
Role	**Backup role**	**Right**	**Capability**	
IAM Service Manager		Read access to OIM Report		

Table G.11: Responsibility *OIM 11*

The task "Auto provisioning and daily reconciliation" is concerned by two accountabilities: the accountability to do the task and the accountability to decide about its achievement. Francis Carambino is assigned to the realisation of this task and he has for backup: Philippe Melvine. Marco Jonhson has to decide about the realisation of the task and he has, therefore, the right to Read OIM Report (Tables G.12 and G.13).

Responsibility *OIM 12*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Do	Auto provisioning and daily reconciliation	Francis Carambino	Marco Jonhson	Philippe Melvine
Role	**Backup role**	**Right**	**Capability**	
OIM Administrator	IAM Service Manager	Read–Write access to OIM tool	OIM administration training	

Table G.12: Responsibility *OIM 12*

[1]Identity and Access Management

Responsibility *OIM 13*				
Accountability	**Task**	**Employee**	**Accountable towards**	**Backup**
Decide about	Auto provisioning and daily reconciliation	Marco Jonhson		
Role	**Backup role**	**Right**	**Capability**	
IAM Service Manager		Read access to OIM Report		

Table G.13: Responsibility *OIM 13*

G5 Results analysis

The instantiation of the responsibilities, after the mapping of the Responsibility metamodel with the ECA metamodel brings the following results:

- **Refinement of the accountabilities of the employees regarding the tasks**. Before the case study, the description of the process according to the ECA metamodel only listed the roles responsible for performing the task. As a result, this description was not accurate enough to know which employees perform which tasks, and which employees decided, give advice, and so forth. E.g., Marco Jonhson did not appear in the process description, although, he is IAM Service Manager (Table G.11 and Table G.13).

  The description of the process, according to the ECA metamodel integrated with the Responsibility metamodel, gives a clear view on all the accountabilities and their assignments to the employees.

- **Formalisation of capabilities required by the employees to perform the accountabilities**. Before the case study, the description of the process did not address the employees' capabilities necessary to perform accountabilities. Employees were assigned to responsibilities without previously knowing if they were capable of assuming them.

  The description of the process, according to the ECA metamodel integrated with the Responsibility metamodel, clearly highlights the capabilities necessary to perform the tasks.

I.e., To Complete Sysper2, Samantha Doherty needed a Sysper2 and SQL training so anyone assigned to this responsibility requires the same training (Table G.3).

- **Formalisation of the rights and access rights required by the employees to perform the accountabilities**. Another difference, in the process description after the case study, is that the rights, and more specifically the access rights, needed to perform an accountability, are clearly listed. I.e., to complete Sysper2, it is necessary to have the access rights to Read–Write and Modify all Sysper2 functions and the rights to use RETO (Table G.3).

- **Association of employee to responsibilities or to roles**. The final improvement is the possibility to assign a task, either to a role or a responsibility. This possibility offers more flexibilities and it reduces the risk of providing access rights to employees who do not need them. I.e., all employees with the role of Human Resource Directorate/RCD are assigned to the responsibility to Release "Note d'information" (Table G.1), although, only one employee advises about the attribution of offices (Table G.7).

G6 Evaluation of the case study

To evaluate the case study, we provided Chapters 4 and Appendix G to François Vernadat which evaluated and provided, during a closing meeting, a pertinent feed–back, as well as his personal feelings about the research.

Globally, the metamodel has been evaluated as good quality and composed of a rich set of concepts. The case study has allowed to evaluate the usage of ReMMo in order to define responsibilities for the user provisioning and the User Provisioning and User Account Management process.

François Vernadat, however, makes two observations. The first one is that it could have been interesting to have the metamodel presented as an ontology. Having such an ontology would have allowed further rigorous evaluation and would have increased the semantic richness of the concepts and of the connections between each of them. The second observation is that the completeness of the metamodel allows, in the meantime, the modelling of many elements, like the rights to use, the capabilities, the accountabilities, and so forth, but it also renders the metamodel sometimes difficult to exploit in practice. Both observations lead to the conclusion that research is often confronted with dilemmas. Whether the researcher tries to bring additional semantics and from there, brings more rigour to the elaboration of an artefact, or the researcher tries to simplify the solution to make it more pragmatic and more easily usable by the practitioner.

G7 Conclusions

In this appendix, we have presented a case study that illustrates how the Responsibility metamodel can be integrated with an existing professional enterprise architecture model to help defining the responsibilities of the employees involved in a business process. Therefore, we have

integrated the Responsibility metamodel with the enterprise architecture metamodel of the European Court of Auditor (the ECA metamodel) and we have instantiated it to model the responsibilities of the employees involved in the Users' Accounts Management process. This process has been considered, according to the ECA metamodel, as a business process for the IT business.

The case study has been realised in three phases:

Firstly, we have presented in Section G2 the integration of the Responsibility metamodel with the ECA metamodel. This integration has been realised following the method presented in Petit (2003). We have translated the ECA metamodel in an UML model, afterwards, and we have analysed the correspondences between the concepts from each metamodels and finally, we have integrated both metamodels in a single one.

To improve the access rights management activity, the court was running a project to enhance the users' accounts management. This enhancement concerns an automation of the provisioning of the OIM tool with COMREF. In the second phase of the case study, we have presented the users' accounts management process associated to the exploitation of this OIM tool and we have worked on renewing it. This phase has contributed to understand and improve the process and prepare the work in phase three.

Thirdly, we have modelled the responsibilities of all the employees involved in the process. Therefore, we have instantiated, using a four steps approach, the concepts from the integrated metamodel elaborated in the first phase. The output of the last phase was a set of thirteen responsibilities. These responsibilities have permitted to improve the description of the task that composes the process and, as a result, have improved the definition of the access rights required by the employees or the roles that are assigned to the accountability concerning the tasks.

After this first evaluation of the Responsibility metamodel, we have decided (1) to complete the Responsibility metamodel with the concepts of sanction, actor, governance rule and source, (2) to define different types of tasks (business, structural, approve, supervise, advice, control, report and support) and (3) to clarify the associations between some metamodel's concepts.

Publication related to this appendix:

- M. Petit, C. Feltus, F. Vernadat, Enterprise Architecture Enhanced with Responsibility to Manage Access Rights – Case Study in an EU Institution, in *Proceedings of The Practice of Enterprise Modeling – 5th IFIP WG 8.1 Working Conference (PoEM)*, Rostock, Germany. 2012.